DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

A NON-LINEAR SIMULATION FOR AN AUTONOMOUS UNMANNED AIR VEHICLE

by

David R. Kuechenmeister
September, 1993

Thesis Advisor: Isaac I. Kaminer
Thesis Co-Advisor: Richard M. Howard

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

EPORT SECURITY CLASSIFICATION NCLASSIFIED	1b. RESTRICTIVE MARKINGS
ECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
ECLASSIFICATION/DOWNGRADING SCHEDULE	
RFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
AME OF PERFORMING ORGANIZATION aval Postgraduate School	6b. OFFICE SYMBOL *(if applicable)* 55
	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
DDRESS *(City, State, and ZIP Code)* onterey, CA 93943	7b. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943
AME OF FUNDING/SPONSORING RGANIZATION	8b. OFFICE SYMBOL *(if applicable)*
	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
DDRESS *(City, State, and ZIP Code)*	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

TLE *(Include Security Classification)*

A NON-LINEAR SIMULATION FOR AN AUTONOMOUS UNMANNED AIR VEHICLE

ERSONAL AUTHOR(S)

Kuechenmeister, David Robert

TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Master's Thesis	FROM _____ TO _____.	September, 1993	133

UPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the al policy or position of the Department of Defense or the United States Government.

COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
LD	GROUP	SUB-GROUP	Non-Linear, Simulation, Unmanned Air Vehicles, Modeling

BSTRACT *(Continue on reverse if necessary and identify by block number)*

Unmanned Air Vehicles have become increasingly important on the modern battlefield. The restrictive requirement inways and special equipment to take off and land was partially solved by the vertical take off and landing Airborne otely Operated Device, AROD. Work done at the Naval Postgraduate School has modified the AROD to not only and launch vertically, but to fly horizontally for the majority of the mission. To realize these capabilities, as well at of autonomous flight, an accurate computer model was required of both the AROD and the avionics test bed aft, Bluebird, in order to design the control and navigation systems. High fidelity, non-linear equations of motion derived in matrix form that represented any six degree of freedom aircraft model, and were then tailored for use on fic aircraft. Computer modeling of the resulting equations of motion, as well as the sensors used on the aircraft, was using SIMULINK and MATLAB software. The resulting computer model provides a non-linear system of equations, n are easily linearized at any desired flight condition, as required by the proposed control and navigation system n

STRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
NAME OF RESPONSIBLE INDIVIDUAL Isaac I. Kaminer	22b. TELEPHONE *(Include Area Code)* (408) 656-2804 — 22c. OFFICE SYMBOL AA/Ka

RM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Approved for public release; distribution is unlimited

A Non–Linear Simulation for an Autonomous Unmanned Air Vehicle

by

David R. Kuechenmeister
Captain, United States Marine Corps
B.S., The Ohio State University, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

September, 1993

ABSTRACT

Unmanned Air Vehicles have become increasingly important on the modern battlefield. The restrictive requirement for runways and special equipment to take off and land was partially solved by the vertical take off and landing Airborne Remotely Operated Device, AROD. Work done at the Naval Postgraduate School has modified the AROD to not only land and launch vertically, but to fly horizontally for the majority of the mission. To realize these capabilities, as well as that of autonomous flight, an accurate computer model was required of both the AROD and the avionics test bed aircraft, Bluebird, in order to design the control and navigation systems. High fidelity, non-linear equations of motion were derived in matrix form that represented any six degree of freedom aircraft model, and were then tailored for use on specific aircraft. Computer modeling of the resulting equations of motion, as well as the sensors used on the aircraft, was done using SIMULINK and MATLAB software. The resulting computer model provides a non–linear system of equations, which are easily linearized at any desired flight condition, as required by the proposed control and navigation system design.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. IMPORTANCE OF UNMANNED AIR VEHICLES 1

B. UAV RESEARCH USING THE AROD VEHICLE 2

C. REQUIREMENT FOR MODELING 3

II. BACKGROUND 4

A. DESCRIPTION OF AROD 4

B. AROD MODELING 5

C. DESCRIPTION OF BLUEBIRD 8

III. AIRCRAFT EQUATIONS OF MOTION 9

A. NOTATION 9

B. COORDINATE SYSTEMS 10

C. SPATIAL ORIENTATION 11

1. Euler Angles 11

2. Quaternions 15

D. DERIVATION OF EQUATIONS OF MOTION 18

1. Linear Equations 19

2. Angular Equations 20

3. State Equations 21

E. EXTERNAL FORCES AND MOMENTS 22

1. Aerodynamic Forces and Moments 22

2. Other Forces and Moments 25

IV. COMPUTER MODELING OF THE AIRCRAFT EQUATIONS OF MOTION 28

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

A. IMPLEMENTATION OF EQUATIONS 28
1. Procedure 28
2. AROD Equations 29
a. *Control Forces and Moments* 30
b. *Additional Forces and Moments* 31
3. Model Validation 32
a. *Kinematic Equations* 32
b. *Gravitational Forces* 34
c. *Additional Forces and Moments* 36
4. Bluebird Equations of Motion 43
a. *Kinematic Equations* 43
b. *Gravitational Forces* 45
c. *Aerodynamic Forces and Moments* 47
B. VALIDATION OF AN INDEPENDENT CASE 50
V. SENSOR AND ACTUATOR MODELING 56
A. ACCELEROMETER MODELING 56
1. Error Model 59
2. Results and Validation 61
B. RATE GYRO MODELING 64
1. Error Modeling 65
2. Results and Validation 66
C. PITCH, ROLL, AND HEADING SENSOR MODELING 70
1. Error Modeling 70
D. MODELING OF AN ARBITRARY SENSOR PLACEMENT . . . 77
1. Linear Accelerations 77
2. Angular Accelerations 79

VI. CONCLUSIONS AND RECOMMENDATIONS 80
A. CONCLUSIONS 80
B. RECOMMENDATIONS 80
APPENDIX A: MATHEMATICAL PROPERTIES 82
A. CROSS PRODUCT PROPERTIES 82
B. DERIVATIVES OF VECTORS 83
C. EQUATIONS USED FOR LINEARIZATION 84
APPENDIX B: NUMERICAL RESULTS 86
A. AROD RESULTS 86
B. BLUEBIRD NUMERICAL RESULTS 90
C. CESSNA 172 RESULTS 94
APPENDIX C: PROGRAM LISTINGS 96
A. AROD MATLAB ROUTINES 96
1. Main Routine 96
2. Supporting Subroutines 100
3. Data and Initialization Subroutines 102
B. BLUEBIRD MATLAB ROUTINES 106
1. Main Routine 106
2. Supporting Subroutines 113
3. Data and Initialization Subroutines 115
REFERENCES 120
INITIAL DISTRIBUTION LIST 122

LIST OF TABLES

2.1 PHYSICAL CHARACTERISTICS OF AROD 6

2.2 VANE DEFLECTION COMBINATIONS FOR POSITIVE ANGLES 6

2.3 PHYSICAL CHARACTERISTICS OF Bluebird 8

4.1 NON-DIMENSIONAL DERIVATIVE DATA FOR AROD 42

4.2 NON-DIMENSIONAL STABILITY DERIVATIVES 49

4.3 NON-DIMENSIONAL CONTROL DERIVATIVES 50

4.4 COMPARISON OF EIGENVALUES FOR BLUEBIRD 50

4.5 EIGENVALUE COMPARISON FOR CESSNA 172 TEST CASE . . 51

5.1 ACCELEROMETER CHARACTERISTICS 60

5.2 GYRO CHARACTERISTICS 66

5.3 INCLINOMETER AND HEADING SENSOR CHARACTERISTICS 71

LIST OF FIGURES

2.1 Airborne Remotely Operated Device 5

2.2 AROD Direction of Positive Vane Deflections 7

3.1 Relative Position of Coordinate Systems 9

3.2 Z–Y–X Euler Angle Rotation Sequence 12

4.1 Gyroscopic Motion of AROD 33

4.2 Modeling of Gravitational Effects for AROD 34

4.3 Velocity of AROD, Gravitational Effects Included 37

4.4 Thrust vs. RPM for AROD 38

4.5 Thrust and Moment Data for AROD 38

4.6 Moment Due to Vane Deflection 40

4.7 Non–Linear Equations of Motion Model 42

4.8 Rolling Motion For Complete Non–Linear Equations 43

4.9 Block Diagram of Kinematic Equations of Motion 44

4.10 Gravitational Forces Model 46

4.11 Gravitational Effects on Velocity 48

4.12 Full Non–Linear Equations of Motion Model 49

4.13 Comparison of Longitudinal Responses to Step Elevator Input 52

4.14 Comparison of Lateral Responses to Step Rudder Input 53

4.15 Comparison of Lateral Responses to Step Aileron Input 53

4.16 Difference in Analytic and Numerical Results, Step Elevator Input . . 54

4.17 Difference in Analytic and Numerical Results, Step Rudder Input . . 54

4.18 Difference in Analytic and Numerical Results, Step Aileron Input . . 55

5.1 Typical Accelerometer Model 57

5.2 Accelerometer Modeling 58
5.3 Synthesis Model for Accelerometers 59
5.4 Error Model for Accelerometers 61
5.5 Measured and True Acceleration From a Step Elevator Input 62
5.6 Measured and True Acceleration From a Step Aileron Input 63
5.7 Measured and True Acceleration From a Step Rudder Input 63
5.8 Functional Diagram of a Rate Gyro 65
5.9 Rate Gyro Model 66
5.10 Rate Gyro Synthesis Model 67
5.11 Measured and True Angular Rates From a Step Aileron Input 68
5.12 Measured and True Angular Rates From a Step Elevator Input 68
5.13 Measured and True Angular Rates From a Step Rudder Input 69
5.14 Simple Pendulum Inclinometer 71
5.15 Inclinometer Error Modeling 72
5.16 Response to a Step Elevator Input 75
5.17 Response to a Step Rudder Input 75
5.18 Response to Step Aileron Input 76

ACKNOWLEDGMENT

I would like to express my appreciation for the assistance given in the development of this thesis by Dr I. I. Kaminer and by Dr. R. M. Howard. Their boundless patience and professional counsel was invaluable to the sucessful completion of this project.

I. INTRODUCTION

A. IMPORTANCE OF UNMANNED AIR VEHICLES

Unmanned Aerial Vehicles have become increasingly more important, both on the battlefield and in civilian service, since the Ryan Q-2C Firebee target drone introduced the "modern age" of UAVs in 1960 [Ref. Siu 91]. From that time on, military planners have assured that UAVs have the capability to collect intelligence, target enemy positions, gather bomb damage assessment, as well as perform many other tasks. The real benefit in using unmanned aircraft lies in the fact that many missions can be performed deep in enemy territory, all without endangering the lives of pilots, or risking the loss of a much more expensive aircraft. With the recent use of UAVs in Operation Desert Storm, improvements in the current technology are both indicated and desirable.

The most capable UAV in service of the United States Navy and Marine Corps today is the Pioneer Short Range UAV. The system is hampered, though, by the large amount of runway and special equipment needed to launch and land the aircraft. These requirements limit the usefulness of the Pioneer by keeping the aircraft take-off and landing area well away from the areas where the ground forces are operating. This distance then leads to longer transit times to and from the assigned operating area, and thus a shorter time on station. However, what is really required in many instances by the ground forces is an aircraft that can respond quickly to a changing tactical situation. The Airborne Remotely Operated Device, AROD, was an attempt by Sandia National Laboratories [Ref. Wh 87], in response to a requirement by the Naval Ocean Systems Center, NOSC, to respond to these needs.

The United States Marine Corps had set a requirement for a short range, direct support UAV as described in [Ref. MCG 87]:

> " ...to allow the front line commander to see "over the next hill", to a distance of two kilometers ..."

The AROD was designed to be a ducted fan, hovering device carrying a fiber optic data link and on board cameras. AROD testing was canceled [Ref. Sa 89] as the Department of Defense requirements grew, requiring a minimum range of 30 km for all Short Range UAVs. The AROD was incapable of this kind of range, however; the design is still potentially useful.

The Unmanned Air Vehicle Flight Research Lab, UAV FRL, at the Naval Postgraduate School has proposed a solution using the AROD that would satisfy the DoD short range UAV requirements, while maintaining the important capability for vertical take-offs and landings.

B. UAV RESEARCH USING THE AROD VEHICLE

Unmanned Aerial Vehicle research underway at NPS has taken the AROD airframe and fitted it with wings from the Aquila UAV [Ref. Kre 92, Sto 93] in order to give the AROD forward flight capability. The proposed configuration will give the Archytas (an AROD with wings) the ability to take-off and land vertically and then transition to horizontal flight for the mission. This design will explore new technology, driven by the goals established by the UAV Joint Project Office [Ref. DOD 92] of:

- Take off weight under 200 lb
- Carry a 50 lb payload
- Fly at a maximum speed of 150 kts

- Take-off and land in an area 30m by 60m

The vertical flight is accomplished with a powerful ducted fan,which causes a great deal of gyroscopic coupling and torque when producing enough thrust to lift the aircraft. Therefore, in order to achieve stable take-offs and landings, a three-axis autopilot is a necessary feature of the aircraft. Additional capabilities desired in the final version of the Archytas are guidance and navigation systems which will allow autonomous operation, as well as a Global Positioning System aided autoland capability.

C. REQUIREMENT FOR MODELING

Simulation and modeling of the aircraft are essential to the successful design of a control system capable of autonomous flight. The model must be a very high fidelity, non-linear model, that can be easily linearized at any given flight condition. The model should be able to interpolate between data points resulting from wind tunnel testing in order to simulate the highly non-linear transition from vertical to horizontal flight. Moreover, the model must also be capable of including the outputs of the sensors as inputs to the control and navigation system for sensors located at any arbitrary location on the aircraft.

This thesis develops a six degree of freedom model for the AROD in the vertical flight regime, as well as for an aircraft in a fixed wing configuration. This test aircraft, named Bluebird, is used to test the guidance, navigation, and control, GNC, systems in horizontal flight, since there currently is no aerodynamic data available for the Archytas configuration. Use of the Bluebird will provide the capability to design and test a GNC system on a stable aircraft before the first flight on the Archytas.

II. BACKGROUND

A. DESCRIPTION OF AROD

The AROD was designed by the Sandia Research Laboratory in Albuquerque, New Mexico in a project managed by NOSC. The vehicle possessed no flying surfaces and relied solely on powered lift for flight. Control of the aircraft was obtained through the use of four fixed anti-torque vanes and four moveable control vanes positioned in the propeller wash of the duct [Ref. We 88]. The main features of the AROD were vertical take-off and landing, VTOL, flight, lightweight construction, compact size, and minimal support equipment required. However, the AROD required most of the engine output to maintain the powered lift, so very little excess thrust was left for translational flight.

An important aspect of the AROD design was the improvement in static performance provided by the efficiency of the ducted fan design. The addition of the shroud around the three-bladed propeller resulted in increased mass flow through the fan, and thus more static thrust when compared to a conventional propeller configuration [Ref. Kre 92]. The AROD is shown in Figure 2.1 and characteristics of the AROD are tabulated in Table 2.1. The moveable control vanes are all used in combination to exert the desired control forces on AROD. Roll control is achieved by deflecting the four vanes in the same direction, while pitch and yaw control is obtained by deflecting a pair of vanes in the required direction. The numbering of the vanes is shown in Figure 2.2 and the combinations of vane deflections required for positive roll, pitch, and yaw motion are given in Table 2.2.

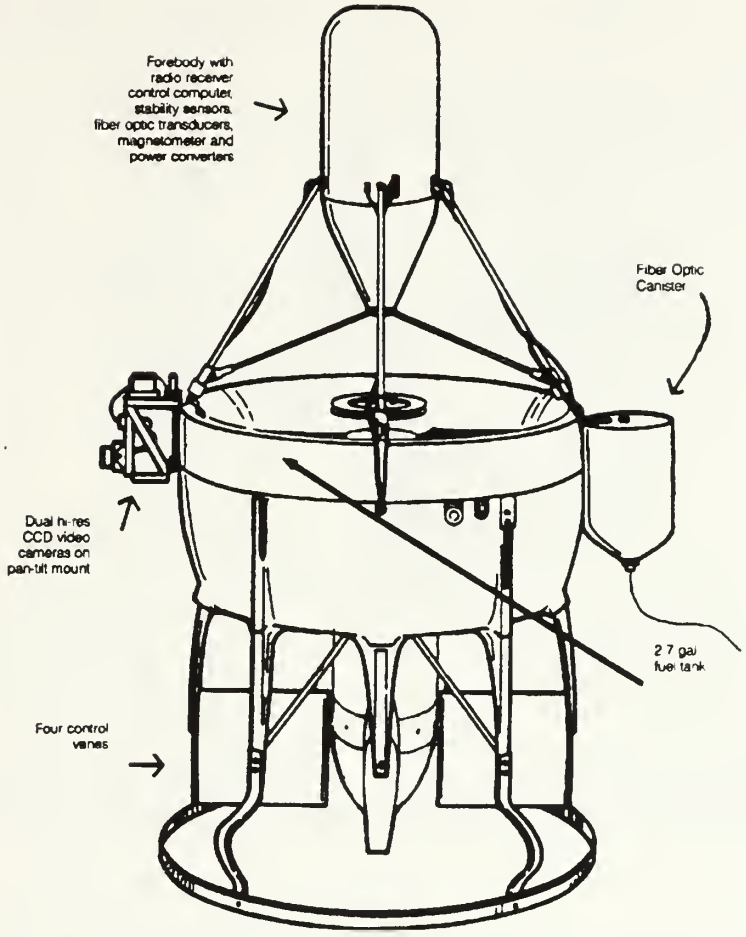

Figure 2.1: Airborne Remotely Operated Device, [Ref. Siu 91]

B. AROD MODELING

The AROD vehicle has been the subject of several theses at NPS. The designs presented in the theses rely on the AROD model given by Sandia Labs in the original design of their controller [Ref. Wh 87, Wh 91]. This model was based on the more classical technique of linearizing an aircraft model, based on dimensional derivatives in a state space form. The resulting model was acceptable for the AROD in a hovering and near vertical translational flight mode, but was not easily adaptable to anything other than the narrow range of conditions planned for AROD. The Sandia Labs papers also pointed out several types of coupling in the AROD. The most prominent

TABLE 2.1: PHYSICAL CHARACTERISTICS OF AROD

Inlet Diameter, A	29.25 in
Propeller Radius, R	12 in
Exit Radius	23.375 in
Inlet Area Ratio	1.219
Exit Area Ratio	1.115
Exterior Contour	Tapered Rear
Propeller Location, % chord	25 %
Number of Blades	3
Engine Speed, Max.	8000 rpm
Engine Speed, Nom.	6500 rpm
Tip Speed, Max.	838 fpm
Tip Speed, Nom.	680 fpm
Power Loading, $\frac{BHP(\rho_0/\rho)}{R^2/4}$	$7.25\ HP/f^2$
Mass Moment of Inertia, I_x	$1.2312\ slug - f^2$
Mass Moment of Inertia, I_y	$3.9548\ slug - f^2$
Mass Moment of Inertia, I_z	$3.9825\ slug - f^2$
Prop Mass Moment of Inertia, I_{rx}	$0.00898\ slug - f^2$
Prop Mass Moment of Inertia, I_{ry}	$0.0045\ slug - f^2$
Prop Mass Moment of Inertia, I_{rz}	$0.0045\ slug - f^2$

TABLE 2.2: VANE DEFLECTION COMBINATIONS FOR POSITIVE ANGLES

	Vane Combination
Roll, Φ	$V_1 + V_2 + V_3 + V_4$
Pitch, Θ	$V_2 - V_4$
Yaw, Ψ	$V_1 - V_3$

of the coupling effects is the gyroscopic coupling between the pitch and yaw axes resulting from the large amount of angular momentum contributed to the aircraft by the propeller. Another dynamic coupling exists between the altitude–rate and the vehicle attitude, since a loss of lift due to thrust will occur when the vehicle is tilted to generate horizontal motion. Yet a third dynamic coupling exists between the altitude

Figure 2.2: AROD Direction of Positive Vane Deflections

and roll control loops, since the reactive torques applied to the roll axis vary as the engine speed is varied. Sandia Labs also provided data for modeling both the engine and the servos as second order transfer functions which were used in this thesis.

Additional information was obtained by Weir [Ref. We 88] in wind tunnel testing. This information included non-dimensional derivatives for vane effectiveness and non-dimensional stability derivatives. The report also stated that the control-vane effectiveness is constant out to at least 25 deg of deflection. Wind tunnel data were also presented to show that control vane effectiveness in approximately the same for translational flight as for hovering flight. This equivalence is due to the fact that the vanes are located in the high speed flow aft of the propeller and are not significantly affected by the freestream.

TABLE 2.3: PHYSICAL CHARACTERISTICS OF BLUEBIRD

Weight	55 lbs
Average Wing Chord, $\bar{c}$	1.802 f
Wing Span, b	12.42 f
Planform Surface Area, S	22.380 f^2
Engine Power	4.0 HP
Mass Moment of Inertia, I_x	10.0 $slug - f^2$
Mass Moment of Inertia, I_y	16.12 $slug - f^2$
Mass Moment of Inertia, I_z	7.97 $slug - f^2$

C. DESCRIPTION OF BLUEBIRD

The Bluebird aircraft was acquired as a test bed for guidance and navigation systems. Ultimately, these systems will be installed on the Archytas aircraft. The Bluebird is a conventional aircraft that will be used to test systems for a similar configuration to the Archytas in forward flight. The aircraft model was developed in the same manner as for AROD, as will be described in Chapter III. Physical characteristics for the Bluebird are given in Table 2.3.

III. AIRCRAFT EQUATIONS OF MOTION

A. NOTATION

In this section the notation used in modeling the equations of motion is introduced. This notation is common in the field of robotics (see [Ref. Sil 91] and [Ref. Cra 86]). and is shown below in Figure 3.1. The following conventions are used:

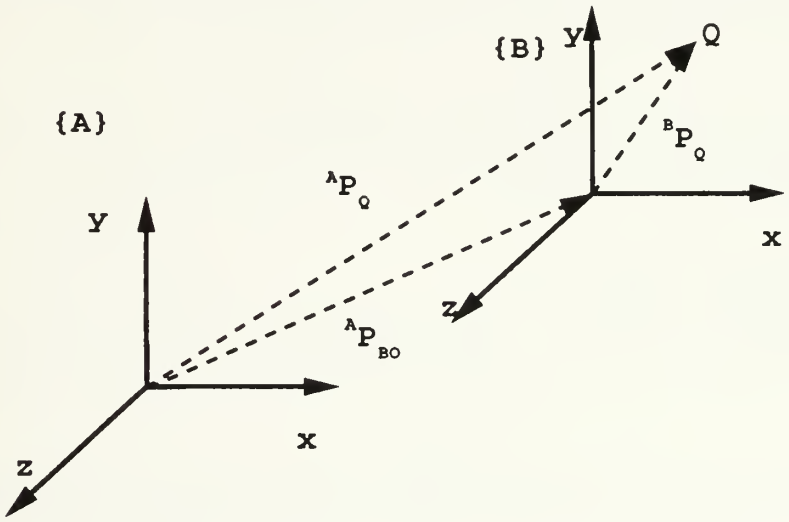

Figure 3.1: Relative Position of Coordinate Systems

- $\{A\}$ represents the coordinate system with basis vectors, x_A, y_A, and z_A.
- ${}^{A}P_Q$ represents the position of point Q, expressed in $\{A\}$.
- ${}^{A}V_Q$ represents the velocity of point Q, measured in $\{A\}$ and expressed in $\{A\}$.
- ${}^{B}({}^{A}V_Q)$ represents the velocity of point Q, measured in $\{A\}$, and expressed in $\{B\}$.
- ${}^{A}_{B}R$ is a rotation matrix, also called a direction cosine matrix. A free vector in one coordinate system, that is a vector that *can be moved parallel to itself*

without change e.g., ${}^{B}V_Q$ can be expressed in another coordinate system by using the rotation matrix:

$$ {}^{A}({}^{B}V_Q) = {}^{A}_{B}R\,({}^{B}V_Q) $$

- ${}^{A}\Omega_B$ is the angular velocity of the $\{B\}$ coordinate system with respect to $\{A\}$, and expressed in $\{A\}$.
- ${}^{B}({}^{A}\Omega_B)$ is the angular velocity of $\{B\}$, with respect to $\{A\}$, and expressed in $\{B\}$.
- Additional information can be added to the subscripts i.e., ${}^{A}P_{BO}$ is the position of the origin of $\{B\}$, expressed in $\{A\}$.

B. COORDINATE SYSTEMS

In order to derive equations of motion for a rigid airplane, the following coordinate systems and assumptions will be used:

- $\{U\}$ represents the inertial tangent plane coordinate system attached to Earth.
- $\{B\}$ represents the body fixed coordinate system.
- All sensors are located at the c.g. (This assumption will be lifted in a later section)
- The mass of the aircraft remains constant.
- Given a vector v, its derivative with respect to $\{B\}$ is denoted as $\frac{d}{dt}(\cdot)$ and its derivative with respect to $\{U\}$ is denoted as $\dot{(\cdot)}$

The $\{B\}$ coordinate system is defined with X_B as the thrust axis. A positive roll rate, p, is clockwise when looking in the positive X direction. The positive direction for Y_B, the pitch axis, is out the right wing . A positive pitch rate, q, is defined as clockwise looking in the positive Y direction. The Z_B axis is the yaw axis, and a positive yaw rate, r, is defined as clockwise, looking in the positive Z direction.

To simplify the notation in places where it becomes cumbersome, The following definitions are introduced:

- v_Q represents the velocity of an arbitrary point, Q, measured and expressed in $\{U\}$.
- v_{BO} represents the velocity of the origin of $\{B\}$, measured and expressed in $\{U\}$, i.e. ${}^{U}V_{BO} = v_{BO}$.
- ${}^{B}v_Q$ represents the velocity of point Q, measured in $\{U\}$ and expressed in $\{B\}$, i.e. ${}^{B}({}^{U}V_Q) = {}^{B}v_Q$.
- ω_B represents the angular velocity of $\{B\}$, measured and expressed in $\{U\}$, i.e. ${}^{U}\Omega_B = \omega_B$.
- ${}^{B}\omega_B$ represents the angular velocity of $\{B\}$ measured in $\{U\}$, and expressed in $\{B\}$, i.e. ${}^{B}({}^{U}\Omega_B) = {}^{B}\omega_B$.

C. SPATIAL ORIENTATION

1. Euler Angles

The spatial orientation of a rigid body [Ref. Ju 92] can be defined by the three Euler angles, Φ, Θ, and Ψ called roll, pitch and yaw and defined in Figure 3.2. The Euler angles,in turn, can be used to define a rotation between two coordinate

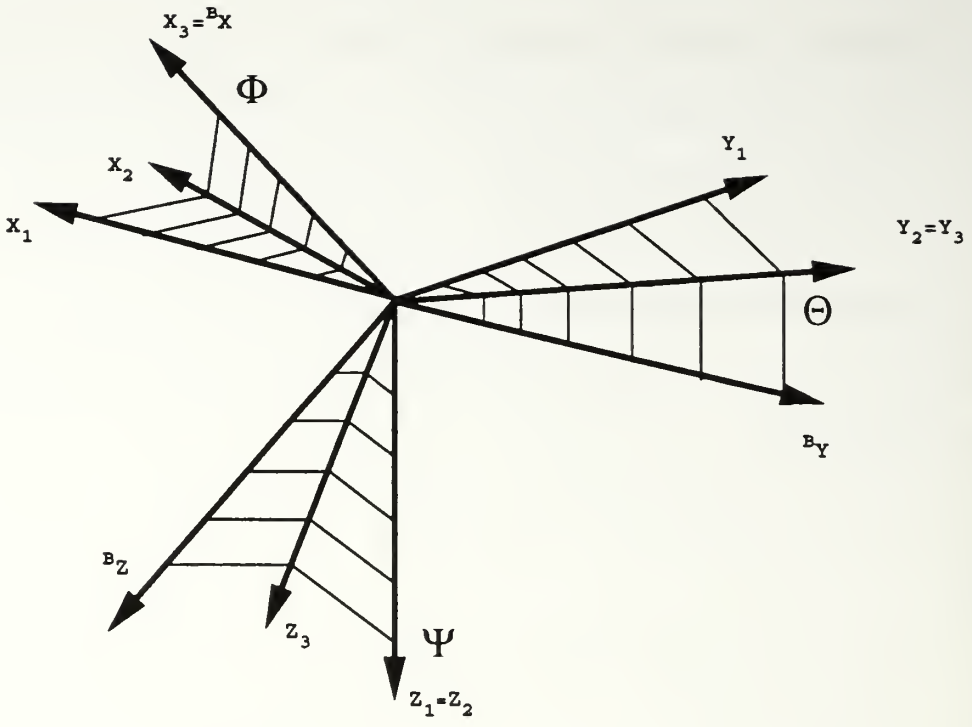

Figure 3.2: Z–Y–X Euler Angle Rotation Sequence

systems. This rotation is obtained using Euler's theorem:

> Any number of rotations about different axes through a point must, in the end, remain equivalent to a single rotation.

For the case of conventional aircraft, a 3–2–1 rotation sequence is used [Ref. Sch 92], where the aircraft is yawed, pitched, then rolled. In the cases investigated here, Θ is small, and in steady state flight is equal to the angle of attack, α. Φ can be expected to be anywhere from $\pm$ 60deg in normal flight and can be anywhere from $\pm$ 180deg in acrobatic flight. Ψ represents the heading of the aircraft and of course can range from 0 to 360 deg. The transformation from inertial coordinates$\{U\}$, to body coordinates $\{B\}$, is carried out as follows, and is shown in Figure 3.2.

1. The inertial coordinate system is represented by the vector ${}^UV$, with the components x, y, and z. The first rotation is made about the z axis through an angle

Ψ. Now the vector is expressed as ${}^2V$ with the components x_2, y_2, and z_2. Since the rotation was about the z axis, the z_2 component remains unchanged. The resulting elemental matrix is:

$$M(\Psi) = \begin{bmatrix} \cos\Psi & \sin\Psi & 0 \\ -\sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.1}$$

2. Now the rotation is made about the new y axis, y_2, through an angle Θ. This results in a third coordinate system with the vector expressed as ${}^3V$, and having components x_3, y_3, and z_3. This rotation does not change the y_3 component. The resulting elemental matrix is:

$$M(\Theta) = \begin{bmatrix} \cos\Theta & 0 & -\sin\Theta \\ 0 & 1 & 0 \\ \sin\Theta & 0 & \cos\Theta \end{bmatrix}. \tag{3.2}$$

3. Lastly, the rotation about the x_3 axis through an angle Φ is made to give the vector expressed in body coordinates, ${}^BV$. Now the resulting matrix is

$$M(\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{bmatrix}. \tag{3.3}$$

When the matrices are multiplied together in the correct sequence, $M(\Phi)M(\Theta)M(\Psi)$, the result is the ${}^B_U R$ direction cosine matrix, expressed in terms of Euler angles as shown

$$\begin{bmatrix} \cos\Psi\cos\Theta & \sin\Psi\cos\Theta & -\sin\Theta \\ \cos\Psi\sin\Theta\sin\Phi - \sin\Psi\cos\Phi & \sin\Theta\sin\Phi\sin\Psi + \cos\Psi\cos\Phi & \cos\Theta\sin\Phi \\ \cos\Psi\sin\Theta\cos\Phi + \sin\Psi\sin\Phi & \sin\Theta\cos\Phi\sin\Psi - \cos\Psi\sin\Phi & \cos\Theta\cos\Phi \end{bmatrix}. \tag{3.4}$$

The next step is to develop the kinematic differential equations that describe the change in Euler angles with time. Following the method used in [Ref. Sch 92], the matrix of differential equations, Ω, can be written as a sum of individual Euler angle rates:

$$\Omega = M(\Phi)\begin{bmatrix} \frac{d}{dt}\Phi \\ 0 \\ 0 \end{bmatrix} + M(\Phi)M(\Theta)\begin{bmatrix} 0 \\ \frac{d}{dt}\Theta \\ 0 \end{bmatrix} + M(\Phi)M(\Theta)M(\Psi)\begin{bmatrix} 0 \\ 0 \\ \frac{d}{dt}\Psi \end{bmatrix}. \tag{3.5}$$

When the matrix algebra in Equation 3.5 is done, the resulting kinematic differential equations for p,q,and r are given as:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\Theta \\ 0 & \cos\Phi & \cos\Theta\sin\Phi \\ 0 & -\sin\Phi & \cos\Theta\cos\Phi \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} \tag{3.6}$$

The matrix on the right hand side of Equation 3.6 is invertible for all $\Theta \neq \pi/2$, and can be used to solve for the Euler angle rates, $\dot{\Phi}, \dot{\Theta}$ and $\dot{\Psi}$:

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi\sec\Theta & \cos\Phi\sec\Theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{3.7}$$

By integrating Equation 3.7, the time history of the Euler angles can be obtained.

The Euler angle method has one drawback. In the kinematic differential equations derived, a singularity occurs for some particular value of either Φ, Θ, or Ψ. In Equation 3.7, this singularity occurs at $\Theta = \pm\pi/2$, which means the that the coordinate transformation is useful for an aircraft in horizontal flight, but useless for an aircraft which requires a vertical take-off or extended periods of vertical flight. Thus, another type of transformation is necessary for aircraft that spend considerable time flying at conditions near $\Theta = \pi/2$.

The first alternative is simply to use another one of the 12 possible Euler angle transformations. It has been shown that the rotation matrix, R, is made up of sequential rotations and can be characterized as the product of three individual matrices where

$$R = M_\gamma(\theta_3)\, M_\beta(\theta_2)\, M_\alpha(\theta_1), \tag{3.8}$$

where the rotation sequence (α, β, γ) represents one of the combinations of integers

$$(1,2,3),(1,3,2),(1,2,1),(1,3,1)$$

$$(2,1,3),(2,3,1),(2,1,2),(2,3,2)$$

$$(3,2,1),(3,1,2),(3,2,3),(3,1,3)$$

and the $M_i(\theta_j)$ are the rotation matrices

$$
\begin{aligned}
M_1(\theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \\
M_2(\theta) &= \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \\
M_3(\theta) &= \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.
\end{aligned}
\tag{3.9}
$$

One can see that by substituting in (3,2,1), and Φ, Θ, and Ψ for θ_1, θ_2, and θ_3 respectively, the matrix shown in Equation 3.4 is obtained. Euler angle transformation matrices for each of these combinations have been calculated and are tabulated in [Ref. Ka 83]. The best Euler angle rotation sequence for an aircraft with flight at $\Theta = \pi/2$ was selected as a 2–3–1, or Y–Z–X, sequence. This sequence will allow flight at $\Theta = \pi/2$ with no corresponding singularity in the kinematic differential equations. Note that this matrix is invertible for $\Psi \neq \pi/2$. The 2–3–1 rotation matrix, R, is described in terms of Euler angles as

$$
\begin{bmatrix} \cos\Theta\cos\Psi & \sin\Psi & -\sin\Theta\cos\Psi \\ -\cos\Theta\sin\Psi\cos\Phi + \sin\Phi\sin\Theta & \cos\Psi\cos\Phi & \sin\Theta\sin\Psi\cos\Phi + \sin\Phi\cos\Theta \\ \cos\Theta\sin\Psi\sin\Phi + \cos\Phi\sin\theta & -\cos\Psi\sin\Theta & -\sin\Theta\sin\Psi\sin\Phi + \cos\Theta\cos\Phi \end{bmatrix}
\tag{3.10}
$$

The kinematic differential equations can be found in [Ref. Ka 83] as

$$
\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \frac{1}{\cos\Psi} \begin{bmatrix} 1 & -\cos\Phi\sin\Psi & \sin\Phi\sin\Psi \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi\cos\Psi & \cos\Phi\cos\Psi \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}
\tag{3.11}
$$

These equations can now be integrated to find the time history of the Euler angles.

2. Quaternions

Another choice for the expression of a body's spatial orientation is the use of quaternions. Quaternions eliminate the disadvantage of the singularity in the second rotation that is associated with the Euler angles. Quaternions have been in

use for quite some time, having been discovered by Euler in a search for complex numbers [Ref. Mo 84]. A quaternion is defined as [Ref. Mo 84]:

$$q \triangleq \beta_0 + i\beta_1 + j\beta_2 + k\beta_3, \tag{3.12}$$

where the parameters are represented by various authors as S, a, b, c by [Ref. Ro 58], χ, ξ, η, and ζ by [Ref. Whi 59], and q_4, q_1, q_2, and q_3 by [Ref. Sil 91].

The components $\beta_0, \beta_1, \beta_2,$ and β_3 are real numbers and the terms $i, j,$ and k are defined in the typical manner for complex numbers, where

$$\begin{array}{ll} i^2 = -1 & ij = -ji = k \\ j^2 = -1 & jk = -kj = i \\ k^2 = -1 & ki = -ik = j \end{array}$$

The norm of a quaternion, q^*q, is required to be 1:

$$q\,q^* = q^*\,q = \beta_0^2 + \beta_1^2 + \beta_2^2 + \beta_3^2 = 1,$$

since $q^* = \beta_0 - i\beta_1 - j\beta_2 - k\beta_3$.

It can be shown that R can be represented as follows:

$$\overset{B}{U}R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \tag{3.13}$$

where

$$\begin{array}{rcl} r_{11} & = & \beta_0^2 + \beta_1^2 - \beta_2^2 - \beta_3^2 \\ r_{12} & = & 2(\beta_1\beta_2 + \beta_0\beta_3) \\ r_{13} & = & 2(\beta_1\beta_3 - \beta_0\beta_2) \\ r_{21} & = & 2(\beta_1\beta_2 - \beta_0\beta_3) \\ r_{22} & = & \beta_0^2 - \beta_1^2 + \beta_2^2 - \beta_3^2 \\ r_{23} & = & 2(\beta_2\beta_3 + \beta_0\beta_1) \\ r_{31} & = & 2(\beta_1\beta_3 + \beta_0\beta_2) \\ r_{32} & = & 2(\beta_2\beta_3 - \beta_0\beta_1) \\ r_{33} & = & \beta_0^2 - \beta_1^2 - \beta_2^2 + \beta_3^2 \end{array} \tag{3.14}$$

All that is required now is to determine the kinematic differential equations using the quaternions.

By expressing subsequent rotations from one coordinate system to another, where β' orients F_1 to F, and β'' orients F_2 to F_1, an algebraic approach [Ref. Mo 84] can be used to relate F_2 to F.

$$\begin{array}{rcl} R(\beta) & = & R(\beta'')R(\beta') \\ R_{ij}(\beta) & = & \sum_{k=1}^{3} R_{ik}(\beta'')R_{kj}(\beta') \end{array} . \tag{3.15}$$

Now the β's can be expressed in terms of β''s and β'''s with the following result

$$\beta = R(\beta'')\beta', \tag{3.16}$$

where

$$R(\beta) = \begin{bmatrix} \beta_0 & -\beta_1 & -\beta_2 & -\beta_3 \\ \beta_1 & \beta_0 & \beta_3 & -\beta_2 \\ \beta_2 & -\beta_3 & \beta_0 & \beta_1 \\ \beta_3 & \beta_2 & -\beta_1 & \beta_0 \end{bmatrix} . \tag{3.17}$$

By regarding the second rotation in Equation 3.16as infinitesimal, the following result is obtained

$$\dot{\beta} = \frac{1}{2} R(\omega*)\beta, \tag{3.18}$$

where

$$\dot{\beta} = \begin{bmatrix} \dot{\beta}_0 \\ \dot{\beta}_1 \\ \dot{\beta}_2 \\ \dot{\beta}_3 \end{bmatrix} , \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} \text{ and } \omega* = \begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix} , \tag{3.19}$$

and

$$R(\omega*) = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} . \tag{3.20}$$

Equation 3.18 can be rewritten as

$$\dot{\beta} = \frac{1}{2} R(\beta)\omega*, \tag{3.21}$$

and in this form can be integrated to give the time history of the orientation of the body.

Using quaternions has the following advantages over Euler angles in representing spatial orientation of a rigid body:

- Four states required to express the spatial orientation.
- Requires almost 30 % fewer calculations [Ref. Ro 58], mainly because no non-linear, trigonometric equations need to be calculated.
- No singularities in Equation 3.21 at any body attitude.

D. DERIVATION OF EQUATIONS OF MOTION

The derivation of equations of motion for a general six degree of freedom airplane model can be divided into two parts. The first part is simply the determination of the equations of motion for any rigid body in space. It is dependent only on the linear and angular momenta of the body. The second part is the calculation of aerodynamic, gravitational, and thrust forces on the airplane. These forces are particular to a certain aircraft and in general can be represented by the stability and control derivatives described later in the thesis.

1. Linear Equations

Equations for linear motion can be calculated using Newton's law, $F = m\,a$. Since most of the sensor information available for feedback to the control and navigation systems is available in the $\{B\}$ coordinate system, the terms for linear accelerations, as well as forces and moments, will be expressed in the body coordinate system. First the position of the aircraft c.g. is determined as ${}^{U}P_{BO}$. Then Coriolis' theorem is applied to obtain linear velocities for the aircraft. Coriolis' theorem is then reapplied to derive the expression for linear accelerations. Then

$$ {}^{U}V_{BO} \triangleq {}^{U}\dot{P}_{BO}. \tag{3.22} $$

Both sides of Equation 3.22 are premultiplied by ${}^{B}_{U}R$ to get:

$$ {}^{B}_{U}R\,{}^{U}V_{BO} = {}^{B}_{U}R\,{}^{U}\dot{P}_{BO} $$

or

$$ {}^{B}v_{BO} = {}^{B}\dot{P}_{BO} \tag{3.23} $$

Now consider Coriolis' theorem

$$ \dot{A} = \frac{d}{dt}A + \omega \times A, \tag{3.24} $$

where $\dot{A}$ and $\frac{d}{dt}A$ use the notation for derivatives previously defined in Section A. The term $\omega \times A$ represents the difference between the relative velocity as measured from the rotating and non-rotating axes [Ref. Gre 88].

Equation 3.24 is applied to ${}^{B}v_{BO}$ in Equation 3.23 to get:

$$ {}^{B}\dot{v}_{BO} = \frac{d}{dt}{}^{B}v_{BO} + {}^{B}\omega_{B} \times {}^{B}v_{BO}. \tag{3.25} $$

Newton's law can now be written as

$$ \begin{aligned} {}^{U}F &= m\,{}^{U}a \\ &= m\,\dot{v}_{BO}, \end{aligned} \tag{3.26} $$

where ${}^{U}F$ is the total external force applied to the aircraft c.g. Equation 3.26 is premultiplied by ${}^{B}_{U}R$ to obtain the result:

$$\begin{aligned} {}^{B}F &= m\, {}^{B}_{U}R\, {}^{U}\dot{v}_{BO} \\ &= m\, {}^{B}\dot{v}_{BO}. \end{aligned} \tag{3.27}$$

when ${}^{B}\dot{v}_{BO}$ is substituted into Equation 3.27, the final result for ${}^{B}F$ is

$$\begin{aligned} {}^{B}F &= m\,(\frac{d}{dt}{}^{B}v_{BO} + {}^{B}\omega_{B} \times {}^{B}v_{BO}) \\ &= m\,\frac{d}{dt}{}^{B}v_{BO} + m\, {}^{B}\omega_{B} \times {}^{B}v_{BO}. \end{aligned} \tag{3.28}$$

2. Angular Equations

The equations for angular accelerations are derived using Euler's law for preservation of angular momentum. These equations are also derived for the aircraft c.g. by applying Coriolis' theorem to the equation for Euler's law:

$${}^{U}\dot{L}_{BO} = {}^{U}N_{BO}, \tag{3.29}$$

where ${}^{U}L_{BO}$ is the angular momentum of the aircraft and ${}^{U}N_{BO}$ is the total external moment applied to the aircraft c.g. Euler's law can be rewritten in $\{B\}$ by premultiplying Equation 3.29 by ${}^{B}_{U}R$ to get

$${}^{B}\dot{L}_{BO} = {}^{B}_{U}R\, {}^{U}N_{BO}. \tag{3.30}$$

Using Coriolis' theorem in Equation 3.24, ${}^{B}\dot{L}_{BO}$ can be rewritten as

$${}^{B}\dot{L}_{BO} = \frac{d}{dt}{}^{B}L_{BO} + {}^{B}\omega_{B} \times {}^{B}L_{BO}. \tag{3.31}$$

The angular momentum, ${}^{B}L_{BO}$, is defined as the product of the inertia tensor, I_B, and the body's angular velocity, ${}^{B}\omega_{B}$, or

$${}^{B}L \triangleq I_B\, {}^{B}\omega_{B} + I_R\, {}^{B}\omega_{R},$$

where I_R and ${}^B\omega_R$ are the moment of inertia and the angular velocity of the propeller, respectively. When this term is substituted into Equation 3.31, the result is

$$ {}^B\dot{L}_{BO} = \frac{d}{dt}(I_B{}^B\omega_B + I_R{}^B\omega_R) + {}^B\omega_B \times (I_B{}^B\omega_B + I_R{}^B\omega_R), \tag{3.32} $$

where I_B is the inertia tensor for the aircraft and I_R is the inertia tensor for any significant spinning object on the aircraft, such as a propeller, turbine , or other rotating disk. The term, ${}^B\omega_R$, is the angular velocity of the rotor, expressed in $\{B\}$. We can carry out the differentiation in Equation 3.32 to get

$$ {}^B\dot{L}_{BO} = I_B\frac{d}{dt}{}^B\omega_B + I_R\frac{d}{dt}{}^B\omega_R + {}^B\omega_B \times (I_B{}^B\omega_B + I_R{}^B\omega_R). \tag{3.33} $$

However, since $d/dt({}^B\omega_B) = {}^B\dot{\omega}_B$ and $d/dt({}^B\omega_R)$ is assumed to be very small, Equation 3.33 results in

$$ {}^B\dot{L}_{BO} = I_B{}^B\dot{\omega}_B + {}^B\omega_B \times (I_B{}^B\omega_B + I_R{}^B\omega_R) \tag{3.34} $$

Now the result in Equation 3.34 can be substituted into Equation 3.29:

$$ {}^BN_{BO} = I_B{}^B\dot{\omega}_B + {}^B\omega_B \times (I_B{}^B\omega_B + I_R{}^B\omega_R). \tag{3.35} $$

The term $I_R{}^B\omega_R$ can be disregarded if it is insignificant compared to I_B and ${}^B\omega_B$ [Ref. Ros 79].

3. State Equations

In the preceeding sections, kinematic equations for the motion of a rigid body were derived in matrix form. These equations can be assembled into a state space representation of the kinematic equations of motion. First, Equations 3.28 and 3.35 can be written as

$$ \begin{bmatrix} {}^BF \\ {}^BN \end{bmatrix} = \begin{bmatrix} m\,\frac{d}{dt}{}^Bv_{BO} + m\,({}^B\omega_B \times {}^Bv_{BO}) \\ I_B{}^B\dot{\omega}_B + {}^B\omega_B \times (I_B{}^B\omega_B + I_R{}^B\omega_R) \end{bmatrix}. \tag{3.36} $$

Equation 3.36 can be rearranged to yield

$$\frac{d}{dt}\begin{bmatrix} m\,{}^{B}v_{BO} \\ I_B\,{}^{B}\omega_B \end{bmatrix} = \begin{bmatrix} m\,(-{}^{B}\omega_B \times {}^{B}v_{BO}) & +{}^{B}F) \\ -{}^{B}\omega_B \times (I_B\,{}^{B}\omega_B + I_R\,{}^{B}\omega_R) & +{}^{B}N) \end{bmatrix}. \tag{3.37}$$

The terms on the left hand side of Equation 3.37 can be normalized by multiplying by $1/m$ and I_B^{-1}, with the final result of

$$\frac{d}{dt}\begin{bmatrix} {}^{B}v_{BO} \\ {}^{B}\omega_B \end{bmatrix} = \begin{bmatrix} -{}^{B}\omega_B \times {}^{B}v_{BO} & + & \frac{{}^{B}F}{m} \\ -I_B^{-1}\,{}^{B}\omega_B \times (I_B\,{}^{B}\omega_B + I_R\,{}^{B}\omega_R) & + & I_B^{-1}\,{}^{B}N \end{bmatrix}. \tag{3.38}$$

E. EXTERNAL FORCES AND MOMENTS

Section D. gives the equations for kinematic motion, as shown in Equation 3.38, for any rigid body. Now it is necessary to distinguish between the different platforms to be modeled in order to give an accurate representation of the aircraft. This is achieved by computing the actual ${}^{B}F$ and ${}^{B}N$ acting on the aircraft. These forces and moments are those due to gravitational, propulsive, and aerodynamic effects, written as

$$\begin{bmatrix} {}^{B}F \\ {}^{B}N \end{bmatrix} = \begin{bmatrix} {}^{B}F_{GRAV} + {}^{B}F_{PROP} + {}^{B}F_{AERO} \\ {}^{B}N_{PROP} + {}^{B}N_{AERO} \end{bmatrix} \tag{3.39}$$

1. Aerodynamic Forces and Moments

The aerodynamic force and moment terms are determined by using a first-order Taylor series expansion around a given nominal operating point. This operating point is the state chosen to represent the aircraft's flight condition. Each term in the series is a partial derivative of ${}^{B}F$ or ${}^{B}N$ with respect to the aerodynamic variables, u/U, α, β, p, q, r [Ref. Sch 92, Th 89]:

$$F_{AERO} = \delta F_{x'}x' + \delta F_{\dot{x}'}\dot{x}' + \delta F_{\Delta}\Delta + F_0. \tag{3.40}$$

Similarly, moment terms can be written as

$$N_{AERO} = \delta N_{x'}x' + \delta N_{\dot{x}'}\dot{x}' + \delta N_{\Delta}\Delta + N_0. \tag{3.41}$$

where x' is given by

$$x' = [\frac{u}{U}, \beta, \alpha, \frac{pb}{2U}, \frac{qc}{2U}, \frac{rb}{2U}] \tag{3.42}$$

and $\dot{x}'$ is given as

$$\dot{x}' = [\dot{\beta}, \dot{\alpha}]. \tag{3.43}$$

Control inputs are represented by the vector Δ:

$$\Delta = [\delta_e, \delta_r, \delta_a] \tag{3.44}$$

where δ_e, δ_r, and δ_a are the elevator, rudder, and aileron inputs, respectively.

Equations 3.40–3.44 can now be combined as follows:

$$\begin{bmatrix} {}^{W}F_A \\ {}^{W}N_A \end{bmatrix} = \bar{q}\bar{S}\{\frac{\partial C}{\partial x'}x' + \frac{\partial C}{\partial \dot{x}'}\dot{x}' + \frac{\partial C}{\partial \Delta}\Delta + C_{FO}\}, \tag{3.45}$$

where $\bar{q} = 1/2\rho V^2$, $\bar{S} = \text{diag}\{S, S, S, Sb, Sc, Sb\}$, and C is the matrix of non-dimensional stability derivatives differentiated with respect to the terms defined in Equation 3.42, 3.43, or 3.44. $\frac{\partial C}{\partial x'}$ is defined as:

$$\begin{bmatrix} C_{L_U} & C_{L_\beta} & C_{L_\alpha} & C_{L_p} & C_{L_q} & C_{L_r} \\ C_{Y_U} & C_{Y_\beta} & C_{Y_\alpha} & C_{Y_p} & C_{Y_q} & C_{Y_r} \\ C_{D_U} & C_{D_\beta} & C_{D_\alpha} & C_{D_p} & C_{D_q} & C_{D_r} \\ C_{l_U} & C_{l_\beta} & C_{l_\alpha} & C_{l_p} & C_{l_q} & C_{l_r} \\ C_{m_U} & C_{m_\beta} & C_{m_\alpha} & C_{m_p} & C_{m_q} & C_{m_r} \\ C_{n_U} & C_{n_\beta} & C_{n_\alpha} & C_{n_p} & C_{n_q} & C_{n_r} \end{bmatrix}.$$

$\frac{\partial C}{\partial \dot{x}'}$ is very similar to $\frac{\partial C}{\partial x'}$, except that only the $\dot{\alpha}$ and $\dot{\beta}$ terms are normally computed, leaving a 6×2 matrix rather than a square matrix. The term $\frac{\partial C}{\partial \Delta}$ is defined as

$$\begin{bmatrix} C_{L_{\delta e}} & C_{L_{\delta r}} & C_{L_{\delta a}} \\ C_{Y_{\delta e}} & C_{Y_{\delta r}} & C_{Y_{\delta a}} \\ C_{D_{\delta e}} & C_{D_{\delta r}} & C_{D_{\delta a}} \\ C_{l_{\delta e}} & C_{l_{\delta r}} & C_{l_{\delta a}} \\ C_{m_{\delta e}} & C_{m_{\delta r}} & C_{m_{\delta a}} \\ C_{n_{\delta e}} & C_{n_{\delta r}} & C_{n_{\delta a}} \end{bmatrix}.$$

C_{F0} is defined to be the vector of steady state coefficients:

$$C_{F0} = \begin{bmatrix} C_{D0} \\ C_{Y0} \\ C_{L0} \\ C_{l0} \\ C_{m0} \\ C_{n0} \end{bmatrix},$$

representing conditions in trimmed, balanced flight. This definition is similar to the definition used by Roskam [Ref. Ros 79]. In other references, the term C_{F0} can refer to the nominal value of the coefficient at $\alpha = 0$. However, in the Taylor series expansion it is more natural to use the first definition of C_{F0}; therefore, it will be used in the following derivation and modeling. The stability and control derivatives are usually computed in the so–called wind axis coordinate system. The wind axis coordinate system, $\{W\}$, is defined as the coordinate system that results when the x_B axis is aligned into the relative wind. This axis will not be aligned with the body coordinate system since the aircraft flies with an angle of attack, α, and can have a sideslip angle, β. The transformation from $\{W\}$ to $\{B\}$ is performed in the same fashion as the Euler angle transformations mentioned earlier. The rotation matrix, ${}^B_W R$, is a function of α and β, and is expressed as

$${}^B_W R = \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix} \tag{3.46}$$

The rotation from $\{W\}$ to $\{B\}$ is made by premultiplying the force or moment vector by ${}^B_W R$. Additionally, since the lift and drag are defined as positive along the negative z_B and x_B axes, we define F_{AERO} and N_{AERO} as

$$F_{AERO} = \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix} \text{ and } N_{AERO} = \begin{bmatrix} l \\ m \\ n \end{bmatrix}. \tag{3.47}$$

In order to write Equation 3.45 in state space form, state variables must be defined. The most commonly used notation to use for the state vector is to use

$$x = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}. \tag{3.48}$$

However, the terms x' and $\dot{x}'$ in Equations 3.40 and 3.41 cannot be used directly as states. Define

$$\begin{aligned} M' &: x \rightarrow x' \\ \dot{M}' &: \dot{x} \rightarrow \dot{x}' \end{aligned} \tag{3.49}$$

where

$$M' = \text{diag}\{1/V_T, 1/V_T, 1/V_T, b/2V_T, c/2V_T, b/2V_T\}$$

and

$$\dot{M}' = \begin{bmatrix} 0 & c/(2V_T) & 0 & 0 & 0 & 0 \\ 0 & 0 & b/(2V_T) & 0 & 0 & 0 \end{bmatrix}$$

are matrices of appropriate dimensions. The complete expression for ${}^{B}F_{AERO}$ and ${}^{B}N_{AERO}$ can now be written as

$$\begin{bmatrix} {}^{B}F_{AERO} \\ {}^{B}N_{AERO} \end{bmatrix} = \bar{q}\bar{S} \begin{bmatrix} {}^{B}_{W}R & 0 \\ 0 & {}^{B}_{W}R \end{bmatrix} \{C_{F_0} + \frac{\partial C}{\partial x'} M' x + \frac{\partial C}{\partial \dot{x}'} \dot{M}' \dot{x} + \frac{\partial C}{\partial \Delta} \Delta\} \tag{3.50}$$

and can be substituted into Equation 3.38.

2. Other Forces and Moments

In addition to the forces and moments due the aerodynamics of the aircraft, forces and moments due to the propulsive and gravitational forces must be considered. Gravitational forces acting on the aircraft, ${}^{B}F_{GRAV}$, can be found by premultiplying

${}^{U}F_{GRAV}$ by the appropriate rotation matrix, where

$$ {}^{U}F_{GRAV} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}. $$

Then

$$ {}^{B}F_{GRAV} = {}^{B}_{U}R\,{}^{U}F_{GRAV}. \tag{3.51} $$

Propulsive forces and moments, ${}^{B}F_{PROP}$ and ${}^{B}N_{PROP}$, are computed directly in $\{B\}$ and can be expressed as:

$$ {}^{B}F_{PROP} = \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} \tag{3.52} $$

and

$$ {}^{B}N_{PROP} = \begin{bmatrix} T_l \\ T_m \\ T_n \end{bmatrix}, \tag{3.53} $$

where T_i's represent the forces or moments due to thrust. Computation of propulsive forces and moments depends on each particular engine installation, and must be determined for the individual aircraft modeled.

Equations 3.51, 2., and 2. can now be substituted into Equation 3.38:

$$ \frac{d}{dt}\begin{bmatrix} {}^{B}v_{BO} \\ {}^{B}\omega_{B} \end{bmatrix} = \begin{bmatrix} -{}^{B}\omega_{B}\times & 0 \\ 0 & -{}^{B}I_{B}^{-1}({}^{B}\omega_{B}\times({}^{B}I_{B}{}^{B}\omega_{B} + I_{R}{}^{B}\omega_{R})) \end{bmatrix}\begin{bmatrix} {}^{B}v_{BO} \\ {}^{B}\omega_{B} \end{bmatrix} + \begin{bmatrix} \frac{I}{m} & 0 \\ 0 & {}^{B}I_{B}^{-1} \end{bmatrix}\begin{bmatrix} {}^{B}F \\ {}^{B}N \end{bmatrix}, \tag{3.54} $$

where

$$ \begin{bmatrix} {}^{B}F \\ {}^{B}N \end{bmatrix} = \left\{ \begin{bmatrix} {}^{B}F_{GRAV} \\ 0 \end{bmatrix} + \begin{bmatrix} {}^{B}F_{PROP} \\ {}^{B}N_{PROP} \end{bmatrix} + \left\{ \begin{bmatrix} {}^{B}_{W}R & 0 \\ 0 & {}^{B}_{W}R \end{bmatrix} \cdot \bar{q}\bar{S}\left\{ C_{FO} + \frac{\partial C}{\partial x'}M'x + \frac{\partial C}{\partial \dot{x}'}\dot{M}'\dot{x} + \frac{\partial C}{\partial \Delta}\Delta \right\} \right\} \right\}. \tag{3.55} $$

In order to write Equation 3.38 in state space form, the terms associated with $\dot{x}'$ must be collected and moved to the left hand side, along with the other time derivative

terms, ${}^{B}\dot{v}_{BO}$ and ${}^{B}\dot{\omega}_{B}$. Let

$$ {}^{B}_{W}T = \begin{bmatrix} {}^{B}_{W}R & 0 \\ 0 & {}^{B}_{W}R \end{bmatrix} \text{ and } M_I = \begin{bmatrix} m & 0 \\ 0 & {}^{B}I_B \end{bmatrix} $$

then the complete non-linear equations of motion for any aircraft can expressed in state space form as follows [Ref. Th 89]:

$$ \frac{d}{dt}\begin{bmatrix} {}^{B}v_{BO} \\ {}^{B}\omega_B \end{bmatrix} = \chi^{-1}\left\{ \left[\begin{bmatrix} -{}^{B}\omega_B\times & 0 \\ 0 & -{}^{B}I_B^{-1}({}^{B}\omega_B \times ({}^{B}I_B{}^{B}\omega_B + I_R{}^{B}\omega_R)) \end{bmatrix} + M_I^{-1}{}^{B}_{W}T\bar{q}\bar{S}\frac{\partial C_F}{\partial x'}M' \right] \begin{bmatrix} {}^{B}v_{BO} \\ {}^{B}\omega_B \end{bmatrix} + M_I^{-1}\left\{ \begin{bmatrix} {}^{B}F_{GRAV} \\ 0 \end{bmatrix} + \begin{bmatrix} {}^{B}F_{PROP} \\ {}^{B}N_{PROP} \end{bmatrix}\delta_T + {}^{B}_{W}T\bar{q}\bar{S}(C_{F_0} + \frac{\partial C_F}{\partial \Delta}\Delta) \right\} \right], \quad (3.56) $$

$$ {}^{U}\dot{P}_{BO} = {}^{U}_{B}R\,{}^{B}v_{BO}, \quad (3.57) $$

and

$$ \dot{\Lambda} = S(\Lambda)^{B}\omega_B. \quad (3.58) $$

where

$$ \chi = I_6 - M_I^{-1}{}^{B}_{W}T\bar{q}\bar{S}\frac{\partial C_F}{\partial \dot{x}'}\dot{M}' \quad (3.59) $$

P is the position vector of the aircraft, and Λ is the matrix of kinematic differential equations based on Euler angles or quaternions.

IV. COMPUTER MODELING OF THE AIRCRAFT EQUATIONS OF MOTION

A. IMPLEMENTATION OF EQUATIONS

1. Procedure

The AROD/Archytas and the Bluebird aircraft models require different non-linear equations of motion. This difference is due to the unique nature of each aircraft. In the case of the AROD, the angular momentum of the propeller is a major factor to be considered, while the aerodynamics effects in a hover are negligible. The Bluebird on the other hand is a conventional aircraft, and exhibits the opposite characteristics. Angular momentum from the propeller is small and, the aircraft requires the stability and control derivatives associated with aerodynamic flight. All the equations were implemented in a systematic manner following the same general approach for either aircraft model. The commercial products MATLAB and SIMULINK, © 1990–1992 the Mathworks, were chosen for the modeling[1], mainly due to the ease of expressing matrix equations. The model was constructed using the following steps.

- Kinematic equations of motion were coded.
- Gravitational forces, with the direction cosine matrix represented by Euler angles were added to the model
- Stability and control derivatives were included in the model, as well as engine thrust, as appropriate for the aircraft being modeled.

[1]All code is listed in APPENDIX C.

- Engine and actuators were added to the model.
- Sensors for control systems were added to the model.

In the first stage of modeling, analytic linearization was also carried out to verify the computer calculations. This was done by analytically linearizing the matrix formed from the six non-linear equations governing the kinematic motion of the body. Nominal values were substituted in, and the results compared to the trimmed and linearized values obtained from the SIMULINK program[2]. The next step required the addition of gravitational terms, and analytic linearization was still manageable. The linearized matrix now included nine equations and nine states with the Euler angle direction cosine matrix, DCM, and ten equations with ten states for the quaternion DCM. Nominal values for each case were again substituted into the linearized matrix and compared to the linearized model derived from SIMULINK. The inclusion of the stability and control derivatives and the thrust terms presented a problem that was much too cumbersome to linearize analytically. Verification of the data at this stage was accomplished by direct comparison of the dimensional derivatives resulting from numerical linearization of the plant. In the case of the AROD the results were compared with data published by Sandia Labs [Ref. Wh 91]; for the Bluebird, eigenvalues were computed and then compared to eigenvalues obtained by classical analytic methods [Ref. Sch 92] and [Ref. Ros 79].

2. AROD Equations

The AROD differs from the Bluebird primarily in that the aerodynamic forces and moments are negligible while the craft is hovering. The powered lift does present some special problems, the predominant difficulty being the gyroscopic coupling of the spinning propeller. Another important consideration is the moment due

[2]Data is tabulated in APPENDIX B.

to the swirl effect of the air from the propeller striking the vanes mounted aft of the propeller. These forces and moments are computed, then substituted in for ${}^{B}F$ and ${}^{B}N$ in Equation 3.38.

a. *Control Forces and Moments*

All of the applied forces and moments in the AROD are due to the four control vanes mounted aft of the propeller. The vanes can be moved in different combinations, as discussed in later in this chapter, in order to maneuver the AROD. The forces and moments, ${}^{B}F$ and ${}^{B}N$, acting on the AROD are computed from a Taylor series expansion around a nominal hover point. Since aerodynamic terms are negligible,

$$\begin{bmatrix} {}^{B}F_{Control} \\ {}^{B}N_{Control} \end{bmatrix} = \bar{q}_i A R \frac{\partial C}{\partial \Delta} \Delta, \tag{4.1}$$

where

- $\Delta = [\delta_e,\ \delta_r,\ \delta_a]'$
- $\bar{q}_i = 1/2\rho V_i^2$
- V_i is the induced velocity through the propeller [Ref. Pro 90] and $V_i^2 = T/(2A\rho)$.
- A is the inlet area, where $A = 3.14\ f^2$.
- R is the propeller radius, where $R = 1.0\ f^2$.

Notice no aerodynamic forces act on AROD due to the movement of the elevator, rudder, or aileron controls. Again, this is because the model of interest is only designed to fly in a stable hover. The other forces and moments involved in these calculations are due to gravitational and propulsive action, and are described next.

b. *Additional Forces and Moments*

The gravitational force term, ${}^{B}F_{GRAV}$, is computed in the $\{B\}$ coordinate system with the rotation matrix, ${}^{U}_{B}R$. For the AROD, ${}^{U}_{B}R$ is determined by the 2–3–1 Euler angle rotation sequence defined in Equation 3.10 and is written as

$$ {}^{B}F_{GRAV} = {}^{B}_{U}R\,{}^{U}F_{GRAV} \text{ where } {}^{U}F_{GRAV} = \begin{bmatrix} 0 \\ 0 \\ m\,g \end{bmatrix}, \tag{4.2} $$

where

$$ {}^{B}F_{GRAV} = m\,g \begin{bmatrix} -\sin\Theta\ \cos\Psi \\ \sin\Theta\ \sin\Psi\ \cos\Phi + \sin\Phi\ \cos\Theta \\ -\sin\Theta\ \sin\Psi\ \sin\Phi + \cos\Theta\ \cos\Phi \end{bmatrix}. \tag{4.3} $$

The forces and moments due to the propeller thrust were determined experimentally [Ref. Sto 93] and are discussed in a later section. For the AROD, the propulsive force, ${}^{B}F_{PROP}$, is acting completely along the x_B axis

$$ {}^{B}F_{PROP} = \begin{bmatrix} F_{T_X} \\ 0 \\ 0 \end{bmatrix}, \tag{4.4} $$

where F_{T_X} is the total thrust, determined as a function of rpm.

The moment resulting from thrust is ${}^{B}N_{PROP}$ and is given by the vector equation

$$ {}^{B}N_{PROP} = \begin{bmatrix} l_T \\ 0 \\ 0 \end{bmatrix} \tag{4.5} $$

where l_T is the rolling moment due to thrust. This rolling moment is due to the swirl of the air as it leaves the propeller and strikes the control vanes. This term was determined experimentally as a function of thrust, and is also discussed further in a subsequent section.

When all the terms are collected, the total force and moment applied to AROD can be expressed as

$$ \begin{bmatrix} {}^{B}F \\ {}^{B}N \end{bmatrix} = \left\{ \bar{q}AR\frac{\partial C}{\partial \Delta}\Delta + \begin{bmatrix} {}^{B}F_{PROP} \\ {}^{B}N_{PROP} \end{bmatrix} + \begin{bmatrix} {}^{B}F_{GRAV} \\ 0 \end{bmatrix} \right\}. \tag{4.6} $$

The force and moment terms in Equation 4.6 can be substituted into Equation 3.38 with the resulting state space equation given by

$$\frac{d}{dt}\begin{bmatrix} {}^{B}v_{BO} \\ {}^{B}\omega_{B} \end{bmatrix} = \left\{ \begin{bmatrix} -{}^{B}\omega_{B} \times {}^{B}v_{BO} \\ -I_{B}^{-1}\,{}^{B}\omega_{B} \times (I_{B}{}^{B}\omega_{B} + I_{R}{}^{B}\omega_{R}) \end{bmatrix} + \begin{bmatrix} I/m & 0 \\ 0 & I_{B}^{-1} \end{bmatrix} \left(\bar{q}AR\frac{\partial C}{\partial \Delta}\Delta + \begin{bmatrix} {}^{B}F_{PROP} \\ {}^{B}N_{PROP} \end{bmatrix} + \begin{bmatrix} {}^{B}F_{GRAV} \\ 0 \end{bmatrix} \right) \right\}. \quad (4.7)$$

Equation 4.7 can now be written in the state space form and programmed using SIMULINK

$$\frac{d}{dt}\begin{bmatrix} {}^{B}v_{BO} \\ {}^{B}\omega_{B} \end{bmatrix} = \left\{ \begin{bmatrix} -{}^{B}\omega_{B} & 0 \\ 0 & -{}^{B}I_{B}^{-1}({}^{B}\omega_{B} \times ({}^{B}I_{B}{}^{B}\omega_{B} + I_{R}{}^{B}\omega_{R}) \end{bmatrix} \begin{bmatrix} {}^{B}v_{BO} \\ {}^{B}\omega_{B} \end{bmatrix} + \begin{bmatrix} I/m & 0 \\ 0 & I_{B}^{-1} \end{bmatrix} \left(\bar{q}AR\frac{\partial C}{\partial \Delta}\Delta + \begin{bmatrix} {}^{B}F_{PROP} \\ {}^{B}N_{PROP} \end{bmatrix} + \begin{bmatrix} {}^{B}F_{GRAV} \\ 0 \end{bmatrix} \right) \right\}. \quad (4.8)$$

3. Model Validation

a. *Kinematic Equations*

As discussed in the beginning of the chapter, the first step in the AROD model validation was to linearize the non-linear equations

$$\frac{d}{dt}{}^{B}v_{BO} = 1/m(-{}^{B}\omega_{B} \times {}^{B}v_{BO}) \quad (4.9)$$

$$\frac{d}{dt}{}^{B}\omega_{B} = {}^{B}I_{B}^{-1}(-{}^{B}\omega_{B} \times ({}^{B}I_{R}{}^{B}\omega_{R} + {}^{B}I_{B}{}^{B}\omega_{B}), \quad (4.10)$$

with the result given by[3]:

$$\frac{d}{dt}\begin{bmatrix} \delta v \\ \delta\omega \end{bmatrix} = \begin{bmatrix} -\omega_{0}\times & v_{0}\times \\ 0 & -{}^{B}I_{B}^{-1}({}^{B}I_{R}{}^{B}\omega_{R} \times -(\omega_{0}\times){}^{B}I_{B} + {}^{B}I_{B}\omega_{0}\times) \end{bmatrix} \begin{bmatrix} \delta v \\ \delta\omega \end{bmatrix} + \begin{bmatrix} 0 \\ -\omega_{0} \times {}^{B}I_{R} \end{bmatrix} \delta\omega_{R}. \quad (4.11)$$

The nominal values for the AROD hover operating point,

$$x = \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.12)$$

[3]See APPENDIX A for a description of the linearization process

can be substituted into Equation 4.11 to get

$$\frac{d}{dt}\begin{bmatrix} \delta v \\ \delta \omega \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.615 \\ 0 & 0 & 0 & 0 & 1.606 & 0 \end{bmatrix} \begin{bmatrix} \delta v \\ \delta \omega \end{bmatrix} \tag{4.13}$$

where the linearized matrix associated with $\delta\omega_R$ is zero, since $-\omega_0 \times {}^B I_R = 0$. These values match the values obtained by linearizing the model using SIMULINK.

The inertial cross coupling from the propeller is evident from the linearized results. The values of $-1.6152\ sec^{-1}$ and $1.6062\ sec^{-1}$ are defined as pitching moment due to yaw rate, m_r, and yawing moment due to pitch rate, n_q, respectively. The gyroscopic coupling is demonstrated by putting a step moment input of 1 second duration along the y axis and observing the time history of the angular rates q and r as shown in Figure 4.1. The AROD has a tendency to spin in the axis orthogonal to the input torque as shown by the motion r about the z axis.

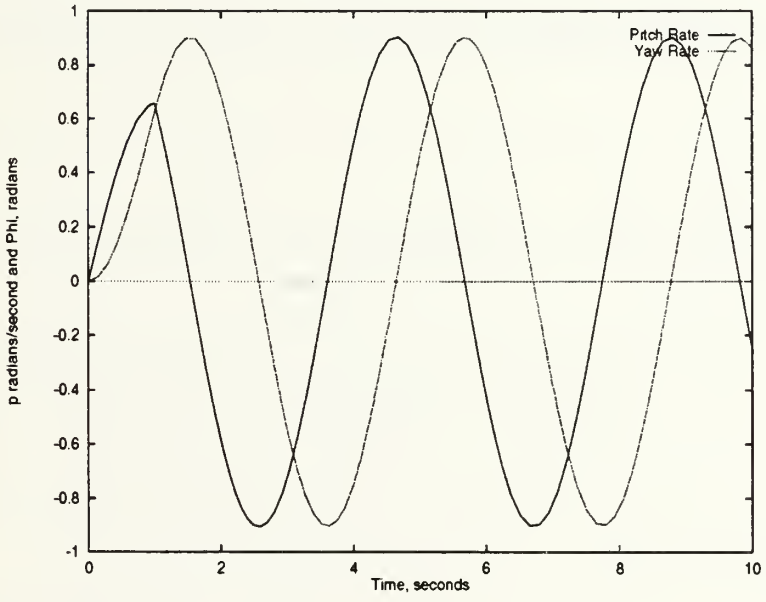

Figure 4.1: Gyroscopic Motion of AROD

b. *Gravitational Forces*

The next step in modeling the AROD was to include gravitational effects into the model. This required the expression for ${}^{B}F_{GRAV}$, given by Equation 4.3, to be coded into a MATLAB function. This function block was then added to the SIMULINK model, as shown in Figure 4.2. The equation to be analyzed at this

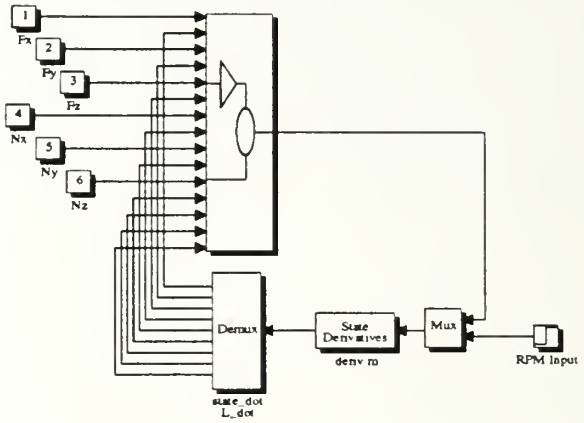

Figure 4.2: Modeling of Gravitational Effects for AROD

stage was

$$\frac{d}{dt}\begin{bmatrix} {}^{B}v_{BO} \\ {}^{B}\omega_{B} \\ \Lambda \end{bmatrix} = \begin{bmatrix} -{}^{B}\omega_{B} \times {}^{B}v_{BO} - \frac{{}^{B}F_{GRAV}}{m} \\ {}^{B}I_{B}^{-1}(-{}^{B}\omega_{B} \times ({}^{B}I_{R}{}^{B}\omega_{R} + {}^{B}I_{B}{}^{B}\omega_{B}) \\ S(\Lambda){}^{B}\omega_{B} \end{bmatrix}, \tag{4.14}$$

where $S(\Lambda)$ represents the kinematic differential equations, defined in Equation 3.11. Notice that since gravity acts through the c.g., no ${}^{B}N_{GRAV}$ term exists. Next, Equation 4.14 was linearized both analytically and with SIMULINK. The analytic result, given in state space form, was

$$\frac{d}{dt}\begin{bmatrix} \delta v \\ \delta\omega \\ \delta\Lambda \end{bmatrix} = \begin{bmatrix} -\omega_0\times & v_0\times & f(\Lambda) \\ 0 & -{}^{B}I_{B}^{-1}({}^{B}I_{R}{}^{B}\omega_{R}\times - (\omega_0\times){}^{B}I_{B} + {}^{B}I_{B}\omega_0\times) & 0 \\ 0 & G(\Lambda) & h(\Lambda,\omega) \end{bmatrix}\begin{bmatrix} \delta v \\ \delta\omega \\ \delta\Lambda \end{bmatrix}$$

$$+\begin{bmatrix} 0 \\ -\omega_0 \times {}^{B}I_R \\ 0 \end{bmatrix} \delta\omega_R. \tag{4.15}$$

In Equation 4.15, the functions $f(\Lambda), G(\Lambda)$, and $h(\Lambda,\omega)$ are partial derivatives of Equations 4.3 and 3.11 with respect to Λ, or $\partial/\partial\Lambda$. The function $f(\Lambda)$ is the linearized expression for ${}^{B}F_{GRAV}$ given by Equation 4.3, where

$$f(\Lambda) = \frac{\partial}{\partial\Lambda}\frac{{}^{B}F_{GRAV}}{m} = g\frac{\partial}{\partial\Lambda}\begin{bmatrix} -\sin\Theta\cos\Psi \\ \sin\Theta\sin\Psi\cos\Phi + \sin\Phi\cos\Theta \\ -\sin\Theta\sin\Psi\sin\Phi + \cos\Theta\cos\Phi \end{bmatrix} \tag{4.16}$$

or

$$g\begin{bmatrix} 0 & -\cos\Theta\cos\Psi & \sin\Theta\sin\Psi \\ -\sin\Theta\sin\Psi\sin\Phi + \cos\Phi\cos\Theta & \cos\Theta\sin\Psi\cos\Phi - \sin\Phi\sin\Theta & \sin\Theta\cos\Psi\cos\Phi \\ -\sin\Phi\cos\Theta - \sin\Theta\sin\Psi\cos\Phi & -\cos\Phi\sin\Theta - \cos\Theta\sin\Psi\sin\Phi & -\sin\Theta\cos\Psi\sin\Phi \end{bmatrix}. \tag{4.17}$$

where $f(\Lambda)$ is evaluated at the nominal condition, Λ_0. The function

$$G(\Lambda) = \partial/\partial\omega(S(\Lambda)^{B}\omega_B)$$

is derived by linearizing the kinematic differential equations in Equation 3.11. The function $h(\Lambda,\omega) = \partial/\partial\Lambda(S(\Lambda)^{B}\omega_B)$ is not presented since for $\omega_0 = 0$, as in steady state cruise, $h(\Lambda,\omega) = 0$. The matrix $G(\Lambda)$ is derived from the following equation

$$G(\Lambda) = \frac{\partial}{\partial\omega}(S(\Lambda)^{B}\omega_B) = \frac{\partial}{\partial\omega}\begin{bmatrix} 1 & -\cos\Phi\tan\Psi & \sin\Phi\tan\Psi \\ 0 & \cos\Phi\sec\Psi & -\sin\Phi\sec\Psi \\ 0 & \sin\Phi & \cos\Phi \end{bmatrix}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.18}$$

$$= \begin{bmatrix} 1 & -\cos\Phi\tan\Psi & \sin\Phi\tan\Psi \\ 0 & \cos\Phi\sec\Psi & -\sin\Phi\sec\Psi \\ 0 & \sin\Phi & \cos\Phi \end{bmatrix}. \tag{4.19}$$

These matrices were assembled as in Equation 4.15 and were evaluated for the nominal flight condition. The values for x_0 for this flight condition were given by

$$x_0 = \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{\pi}{2} \\ 0 \end{bmatrix}. \tag{4.20}$$

After substitution into Equation 4.15, the result for the linearized equations of motion was

$$\frac{d}{dt}\begin{bmatrix} \delta v \\ \delta \omega \\ \delta \Lambda \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 32.174 \\ 0 & 0 & 0 & -10 & 0 & 0 & 0 & -32.174 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.615 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.606 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.21}$$

This result was essentially indentical to the results obtained by using the trim and linearize functions in SIMULINK. The gravitational effects acting on the aircraft were examined by running a simulation of the non-linear model and comparing the results to those from Equation 4.11. The expectation is that a body falling in earth's gravity will experience an acceleration of $32.174 f/s^2$ and as shown in Figure 4.3, this expectation is realized in the model. In Figure 4.3, at the end of the 10 second simulation, the vertical velocity, u in this case, is $\sim 320 f/s$, as was predicted.

c. *Additional Forces and Moments*

The last step in modeling the AROD was to include the forces and moments due to propulsive and control action that act on the aircraft. The data used

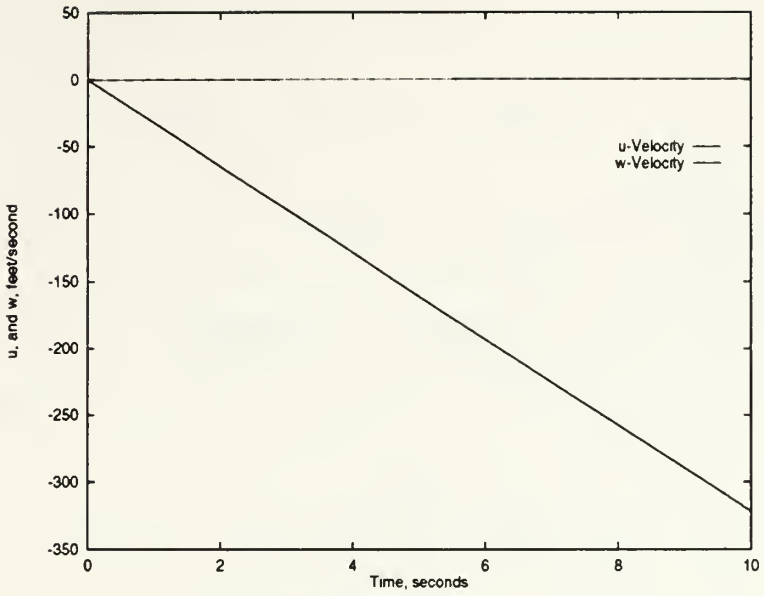

Figure 4.3: Velocity of AROD, Gravitational Effects Included

in modeling these control forces were collected experimentally [Ref. Sto 93] and then curve fitted to a function. The measurements of rolling moment, thrust, rpm, and vane position were taken for various configurations. The data collected were then reduced and the required characteristics were computed. The accuracy of the AROD model depended on very accurate modeling of thrust as a function of rpm, moment as a function of thrust produced, and moment produced by deflecting the control vanes in the different combinations.

Thrust and rolling moments were measured directly at different power settings ranging from 3000 rpm to 7600 rpm, with a power setting of $\sim$ 6400 rpm giving a thrust of $\sim 90 lb_f$. This thrust is aprroximately the force required to maintain a hover for the basic AROD configuration. Figure 4.4 shows the linear curve fit through the thrust vs. rpm data. The curve was fit using data from 5000 rpm to 7600 rpm, as this was expected to approximate the normal operation range in flight. The thrust and moment data are plotted in Figure 4.5 along with the line fit through those data points. The best fit, by least squares, for the data in Figure 4.4 was given

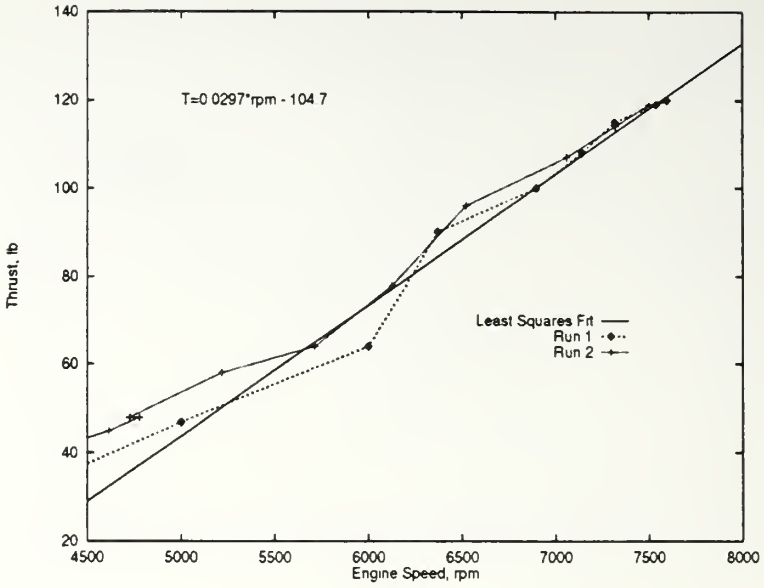

Figure 4.4: Thrust vs. RPM for AROD

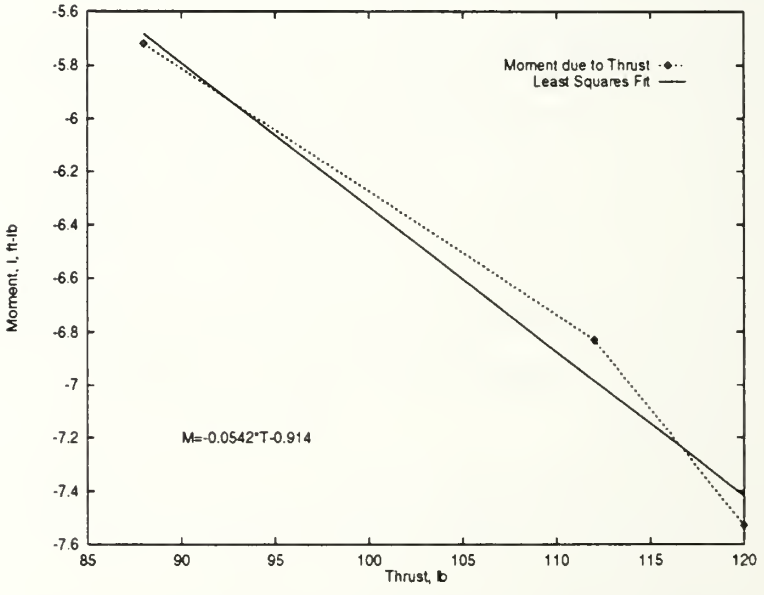

Figure 4.5: Thrust and Moment Data for AROD

by

$$F_{T_x} = 0.0297\,\delta_{rpm} - 104.7, \tag{4.22}$$

where δ_{rpm} represents the rpm at a given throttle setting. The best fit for the data

in Figure 4.5 was given by

$$N_T = -0.0542\, F_{T_x} - 0.9138. \tag{4.23}$$

These equations were then used in modeling the engine response to a rpm setting.

The engine itself was modeled using a second order transfer function from the servo position to δ_{rpm}, based on the Sandia Labs [Ref. Wh 87] model. The transfer function for the engine was given as

$$\delta_{rpm} = \frac{K_E\, \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\delta_T \tag{4.24}$$

where $K_E = 900$, $\omega_n = 5 rad/sec$, $\zeta = 1.0$, and δ_T is the throttle servo position. Since the actual throttle position is set via a radio link and tests have not been set up to model the response, it was ignored in the model.

The engine servo could be modeled and a transfer function from commanded input to servo position was determined. This transfer function also was determined by Sandia Labs in the original AROD work [Ref. Wh 87] to be

$$\delta_T = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}u_T \tag{4.25}$$

where $\omega_n = 20 rad/sec$, $\zeta = 0.6$, and u_T represents the commanded input to the servo. It should be noted that Equation 4.25 was also used for the control vane servos, since all the servos in the AROD were identical.

In order to model the moments due to control surface deflection, ${}^{B}N_{Control}$, computation of control vane effectiveness was necessary. Again, the data gathered by [Ref. Sto 93] was used to compute vane effectiveness for the AROD model. Figure 4.6 shows the moment data for 75% thrust. This power setting corresponds closely to the thrust required to maintain hovering flight. A line was fitted through the data points, and the slope of this line was used to determine the rolling

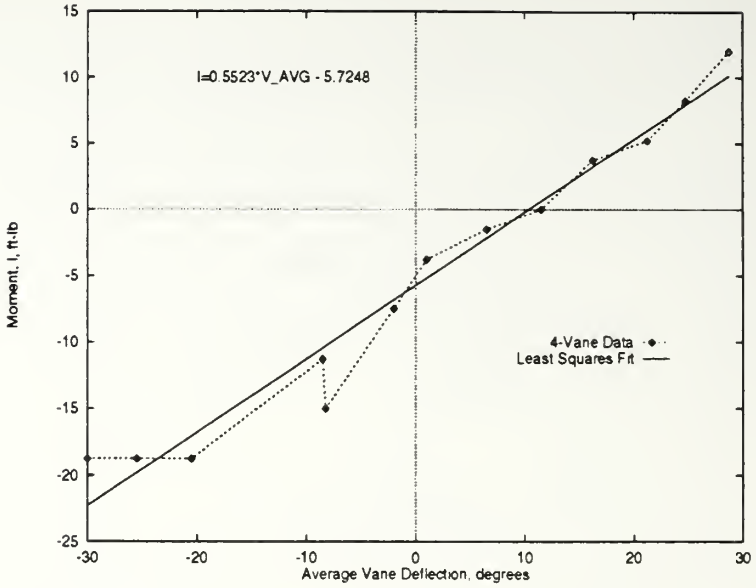

Figure 4.6: Moment Due to Vane Deflection

moment,l, for a given vane deflection. The equation for the curve was

$$l = 0.5523\, V_{AVG} - 5.7248 \tag{4.26}$$

for 6580 rpm, where V_{AVG} was the average vane deflection, $1/4 \sum_{i=1}^{4} V_i$, in degrees. This averaging was required since the individual vanes were at slightly different positions with respect to the commanded position. Computing the vane effectiveness for roll was done by differentiating with respect to the vane deflection,

$$l_{\delta_a} = \frac{\partial l}{\partial V_{AVG}} = 0.5523\ \frac{ft - lb_f}{deg}. \tag{4.27}$$

The rolling moment should be non-dimensionalized for use in the equations of motion given by Equation 4.8. This was done by applying the following equation

$$C_{l_{\delta_a}} = \frac{l_{\delta_a}}{1/2\rho V^2 S b}. \tag{4.28}$$

To use the measurable quantities available, redefining the terms in Equation 4.28 was necessary. The representative area, S, is defined as the inlet area to the duct, A. The

characteristic length, b, is defined as the propeller radius, R. The velocity term was defined as induced velocity, V_i. With these terms the non-dimensional coefficient $C_{l_{\delta_a}}$ could be calculated as

$$C_{l_{\delta_a}} = \frac{l_{\delta_a}}{1/2\rho V_i^2 AR} \tag{4.29}$$

and was computed to be $C_{l_{\delta_a}} = 1.438/rad$. The rolling moment was measured during the testing and was easily non-dimensionalized in a form suited to computer modeling. However, no measured data existed for the pitching and yawing moments. This required the estimation of pitch and yaw vane effectiveness by a simple ratio technique. First, it was assumed that the vane effectiveness, l_{δ_a} for two vanes $(2V)$, would be exactly twice the l_{δ_a} for one vane, and 1/2 the l_{δ_a} for four vanes, $(4V)$. This would make

$$(\frac{\partial l}{\partial V_{AVG}})_{2V} = \frac{1}{2}(\frac{\partial l}{\partial V_{AVG}})_{4V} \tag{4.30}$$

or numerically,

$$(\frac{\partial l}{\partial V_{AVG}})_{2V} = 0.2762\frac{ft - lb_f}{deg} \tag{4.31}$$

Now, the ratio of the dimensional derivatives to the moment arms was set up as

$$\frac{(\frac{\partial l}{\partial V_{AVG}})_{2V}}{\bar{l}_x} = \frac{(\frac{\partial m}{\partial V_{AVG}})}{\bar{l}_y} \tag{4.32}$$

where $\bar{l}_x$ was the distance from the c.g. along the x-axis, to the midspan of the control vane, a distance of 9.0 in. The distance $\bar{l}_y$ was the distance from the c.g. along the y-axis, to the 1/4 chord of the control vane, a distance of 15.43 in. The calculation was performed and the result for the pitching moment was

$$\frac{\partial m}{\partial V_{AVG}} = 0.4735\frac{ft - lb_f}{deg} \tag{4.33}$$

This was non-dimensionalized using Equation 4.28 with the result of $C_{m_{\delta_e}} = 1.2332rad^{-1}$. Moreover because of the symmetrical design of AROD, $C_{m_{\delta_e}} = C_{n_{\delta_r}}$. The results for

vane effectiveness are important to a high fidelity model and are presented along with other relevant data in Table 4.1. The non-dimensional derivatives in Table 4.1

TABLE 4.1: NON-DIMENSIONAL DERIVATIVE DATA FOR AROD

	rad^{-1}	deg^{-1}	Positive Vane Deflection
$C_{l_{\delta_a}}$	1.438	0.0251	$V_1 + V_2 + V_3 + V_4$
$C_{m_{\delta_e}}$	-1.233	-0.0215	$V_4 - V_2$
$C_{n_{\delta_r}}$	-1.233	-0.0215	$V_3 - V_1$

were substituted into the term $(\partial C_F/\partial\Delta)$ in Equation 4.7. Written as a matrix, the non-dimensional derivatives are

$$\frac{\partial C_F}{\partial \Delta}\Delta = \begin{bmatrix} C_{m_{\delta_e}} & 0 & 0 \\ 0 & C_{n_{\delta_r}} & 0 \\ 0 & 0 & C_{l_{\delta_a}} \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_r \\ \delta_a \end{bmatrix} \tag{4.34}$$

Equations 4.22, 4.23, and 4.34 were added as a separate block in the model, resulting in a block diagram shown in Figure 4.7.

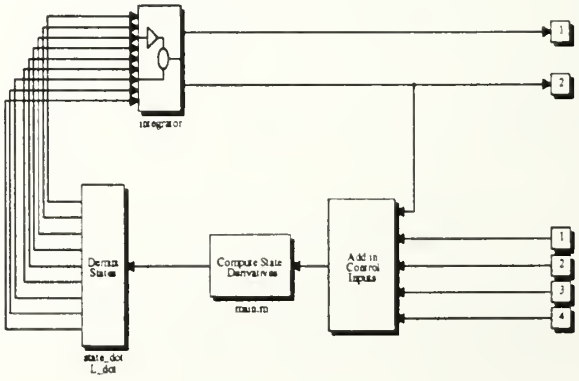

Figure 4.7: Non-Linear Equations of Motion Model

In the SIMULINK linearization results, no change was expected in the A matrix since no aerodynamic terms were added to the model. Rolling motion is shown in Figure 4.8 as an example of the unstable nature of the AROD without a stability augmentation system.

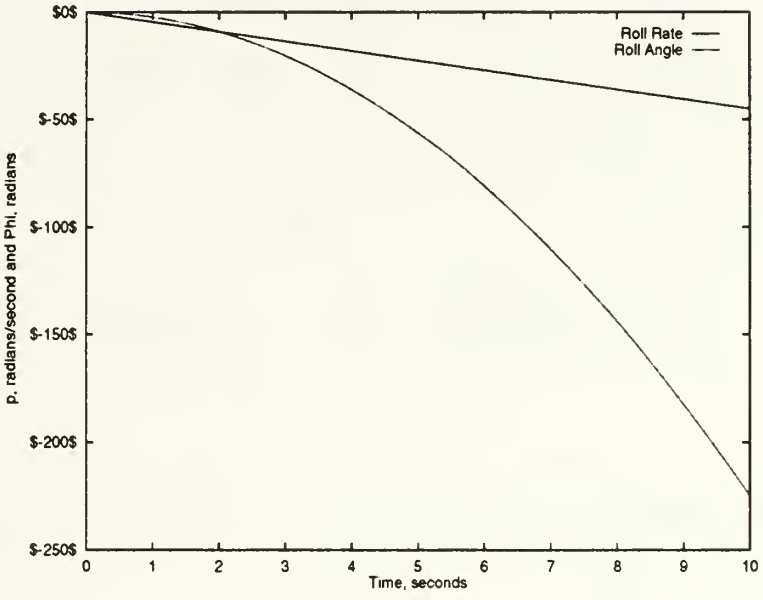

Figure 4.8: Rolling Motion For Complete Non–Linear Equations

4. Bluebird Equations of Motion

a. *Kinematic Equations*

Modeling the kinematic equations of motion for Bluebird was accomplished using the same procedure as was used in the AROD modeling. A slight difference arose in the derivation of the angular acceleration equation for Bluebird since the term, $I_R\omega_R$, for the angular momentum due to propeller rotation is negligible. Thus, the following equations were used in the first stage of modeling.

$$\frac{d}{dt}{}^B v_{BO} = -{}^B\omega_B \times {}^B v_{BO} \tag{4.35}$$

$$\frac{d}{dt}{}^B\omega_B = {}^B I_B^{-1}(-{}^B\omega_B \times {}^B I_B {}^B\omega_B). \tag{4.36}$$

These equations were coded into a MATLAB function and placed into a block diagram, shown in Figure 4.9. Equations 4.35 and 4.36 were linearized analytically[4] with the

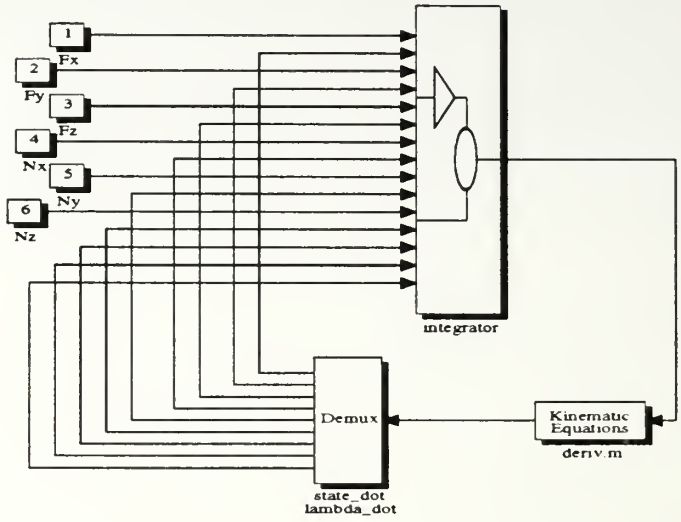

Figure 4.9: Block Diagram of Kinematic Equations of Motion

resulting state space equation

$$\frac{d}{dt}\begin{bmatrix} \delta v \\ \delta \omega \end{bmatrix} = \begin{bmatrix} -\omega_0 \times & v_0 \times \\ 0 & -{}^B I_B^{-1}\{-(\omega_0 \times){}^B I_B + {}^B I_B(\omega_0 \times)\} \end{bmatrix}\begin{bmatrix} \delta v \\ \delta \omega \end{bmatrix}. \tag{4.37}$$

This equation was evaluated at the nominal flight conditions, determined by the typical cruise condition for the Bluebird aircraft. A state vector for the nominal flight conditions is given by

$$x_0 = \begin{bmatrix} 72.9954\ f/s \\ 0 \\ 6.6757\ f/s \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4.38}$$

[4]See APPENDIX A for the details

These values of x_0 were substituted into Equation 4.37:

$$\frac{d}{dt}\begin{bmatrix} \delta v \\ \delta\omega \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & -6.675 & 0 \\ 0 & 0 & 0 & 6.675 & 0 & -72.995 \\ 0 & 0 & 0 & 0 & 72.995 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta v \\ \delta\omega \end{bmatrix} \tag{4.39}$$

These results were in complete agreement with the data obtained by trimming and linearizing the non-linear model with the SIMULINK program. The comparison between Equation 4.39 and Equation 4.13 shows the absence of any angular rate cross coupling in Bluebird. The absence of the cross coupling terms results from the choice to ignore the angular momentum contribution from the propeller.

b. *Gravitational Forces*

After the basic kinematic equations Equation 4.35 and 4.36 were put into block diagram form, it was an easy matter to include additional blocks. The next block was a function block including the ${}^{B}F_{GRAV}$ terms. The model with the gravitational terms included is shown in Figure 4.10. The equations of motion to be modeled at this stage were

$$\frac{d}{dt}{}^{B}v_{BO} = -{}^{B}\omega_B \times {}^{B}v_{BO} - \frac{1}{m}{}^{B}F_{GRAV} \tag{4.40}$$

$$\frac{d}{dt}{}^{B}\omega_B = {}^{B}I_B^{-1}(-{}^{B}\omega_B \times {}^{B}I_B{}^{B}\omega_B) \tag{4.41}$$

$$\frac{d}{dt}\Lambda = S(\Lambda){}^{B}\omega_B \tag{4.42}$$

where $S(\Lambda)$ is the set of kinematic differential equations based on the 3–2–1 Euler angle rotation in Equation 3.7. These equations were then linearized analytically, with the result

$$\frac{d}{dt}\begin{bmatrix} \delta v \\ \delta\omega \\ \delta\Lambda \end{bmatrix} = \begin{bmatrix} -\omega_0\times & v_0\times & g\,f(\Lambda) \\ 0 & -{}^{B}I_B^{-1}\{-(\omega_0\times){}^{B}I_B + {}^{B}I_B(\omega_0\times)\} & 0 \\ 0 & G(\Lambda) & h(\Lambda,\omega) \end{bmatrix} \begin{bmatrix} \delta v \\ \delta\omega \\ \delta\Lambda \end{bmatrix} \tag{4.43}$$

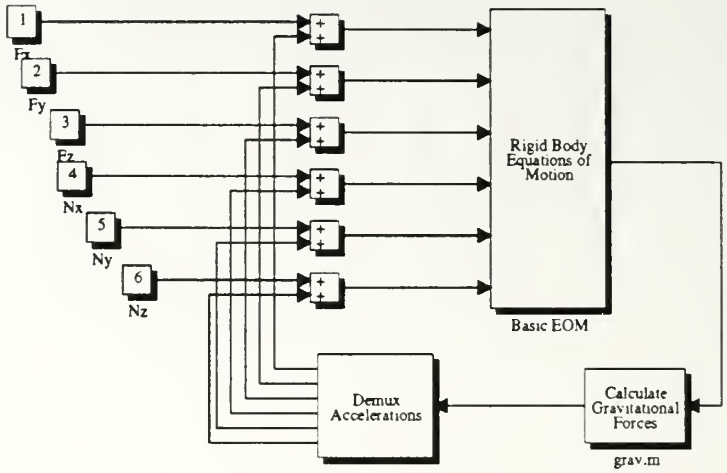

Figure 4.10: Gravitational Forces Model

where the terms $f(\Lambda)$, $G(\Lambda)$, and $h(\Lambda, \omega)$ are based on the 3–2–1 Euler rotation sequence. The linearization is done in the same manner as was shown in Equations 4.16, 4.17, and 4.19 resulting in

$$f(\Lambda) = \begin{bmatrix} 0 & -\cos\Theta & 0 \\ \cos\Theta\cos\Phi & -\sin\Theta\sin\Phi & 0 \\ -\cos\Theta\cos\Phi & -\sin\Theta\cos\Phi & 0 \end{bmatrix}, \tag{4.44}$$

and

$$G(\Lambda) = \begin{bmatrix} 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi\sec\Theta & \cos\Phi\sec\Theta \end{bmatrix}. \tag{4.45}$$

Note that

$$h(\Lambda, \omega) = \frac{\partial}{\partial\Lambda} S(\Lambda)^B\omega_B|_0 = 0,$$

since $\omega_0 = 0$. When Equation 4.43 is evaluated at the nominal condition x_0 given by

$$x_0 = \begin{bmatrix} 72.9954\ f/s \\ 0 \\ 6.6757\ f/s \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.0912\ rad \\ 0 \end{bmatrix},$$

the resulting equation is

$$\frac{d}{dt}\begin{bmatrix} \delta v \\ \delta\omega \\ \delta\Lambda \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & -6.675 & 0 & 0 & -32.043 & 0 \\ 0 & 0 & 0 & 6.675 & 0 & -72.995 & 32.043 & 0 & 0 \\ 0 & 0 & 0 & 0 & 72.995 & 0 & 0 & -2.930 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0.0915 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.004 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta v \\ \delta\omega \\ \delta\Lambda \end{bmatrix}. \tag{4.46}$$

In this instance, as well, the results of the analytic linearization in Equation 4.46 agree very closely with the computed results.

A non-linear simulation of the system in Figure 4.10 should show an increasing downward velocity due to the gravitational effects of ${}^{B}F_{GRAV}$. One would also expect to see decreasing forward velocity due to the "drag-like" term that arises with the introduction of the angle, Θ. This plot is shown in Figure 4.11.

c. *Aerodynamic Forces and Moments*

Completion of the Bluebird equations of motion model required the modeling of the aerodynamic forces and moments acting on the aircraft. A simple engine model was also developed. No analytic linearization was performed at this stage due to the increased complexity of the model. Verification of the computer results was accomplished by comparing the modes and eigenvalues of the computed

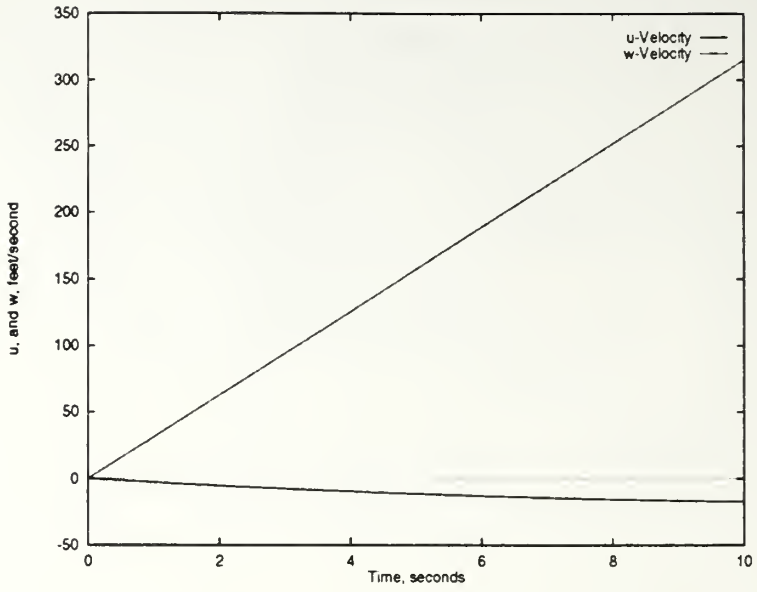

Figure 4.11: Gravitational Effects on Velocity

plant with those resulting from substituting the stability and control derivatives into equations developed in [Ref. Sch 92].

The aerodynamic forces and moments as described in Equation 3.50 were coded as a MATLAB function, then included as a block in the model shown in Figure 4.12.

Next, it was necessary to premultiply all the blocks by χ^{-1}, since this added the effects of the $\dot{\alpha}$ derivatives. Now the important task was to calculate the stability and control derivatives using the general reference for the estimation of non-dimensional derivatives, DATCOM [Ref. USAF 60]. The stability and control derivatives were computed based the aircraft geometry and the control surfaces. These values are tabulated in Table 4.2 and in Table 4.3 where the non-dimensional force is listed on the left side, and the particular derivative is determined using the top row. For example, $C_{D_\alpha} = 0.188$ using Table 4.2. The terms in C_{F_0} were also estimated to be $C_{L_0} = 0.385$ and $C_{D_0} = 0.03$, with all other terms equal to zero since the aircraft is in straight and level flight at this trim condition.

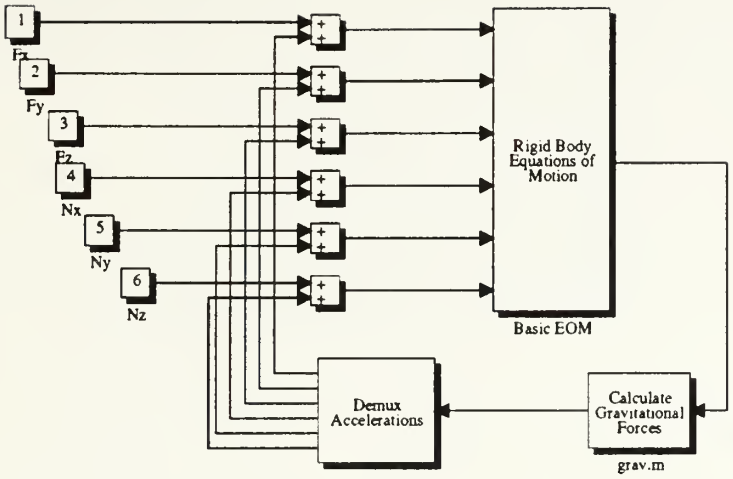

Figure 4.12: Full Non–Linear Equations of Motion Model

TABLE 4.2: NON–DIMENSIONAL STABILITY DERIVATIVES

	u	β	α	p	q	r	$\dot{\alpha}$
C_D	0	0	0.188	0	0	0	0
C_Y	0	-0.31	0	0	0	0.0973	0
C_L	0	0	4.22	0	3.94	0	1.32
C_l	0	-0.0597	0	-0.363	0	0.100	0
C_m	0	0	-1.163	0	-11.77	0	-4.70
C_n	0	0.0487	0	-0.0481	0	-0.0452	0

The ${}^{B}F_{PROP}$ was based on estimating engine thrust for a 4 HP engine and a propeller efficency, η_P, of 0.65. Thrust, T_0 was estimated using the equation

$$T = \frac{550\,\eta_P\,HP}{U_0}\frac{\rho_n}{\rho_0} \tag{4.47}$$

where ρ_n is the density at the operating altitude, ρ_0 is the sea level density, and U_0 is the velocity of the aircraft in f/s. The result was an engine thrust of $T_0 = 19.5lb_f$, for a density ratio of 1. This could be directly factored into the equations of motion as $T_0\delta_T$, where δ_T was arbitrarily scaled from zero to one. $\delta_T = 0$ represents zero thrust

TABLE 4.3: NON–DIMENSIONAL CONTROL DERIVATIVES

	δ_e	δ_r	δ_a
C_D	0.065	0	0
C_Y	0	0.0697	0
C_L	0.472	0	0
C_l	0	0.0028	0.265
C_m	-1.410	0	0
C_n	0	-0.0329	-0.0347

and $\delta_T = 1$ represents maximum thrust.

The preceeding values were substituted into the appropriate MATLAB functions and the entire equations of motion model was trimmed, linearized, and then compared with the analytic results based on classical techniques for determining eigenvalues and eigenvectors. The eigenvalues are compared in Table 4.4. The results from the computed eigenvalues is very close to the eigenvalues derived by analytic methods.

TABLE 4.4: COMPARISON OF EIGENVALUES FOR BLUEBIRD

MODE	COMPUTED	ANALYTIC
LONGITUDINAL		
Phugoid	$-0.0191 \pm 0.4963j$	$-0.0473 \pm 0.4940j$
Short Period	$-3.9833 \pm 3.5521j$	$-4.0034 \pm 3.5462j$
LATERAL		
Dutch Roll	$-0.5285 \pm 3.6346j$	$-0.522 \pm 3.6194j$
Short Period	-5.5629	-5.6654
Spiral	+0.0420	+0.0420

B. VALIDATION OF AN INDEPENDENT CASE

Although verification of the model was accomplished at each stage, comparison of the results from the numerical linearization with linearized results from an inde-

pendent source was still necessary before the model could be considered completely reliable. The test case selected was the Cessna 172 documented in [Ref. Ros 79] as airplane A. The tabulated non-dimensional derivatives and given flight conditions were used as inputs to the non-linear model. The model was then trimmed at the specified flight condition of $V_T = 219\ f/s$ and $\Theta_0 = 0$. Using the resulting state and input vector, the model was linearized around thesed nominal conditions. With the linearized plant, eigenvalues were determined and compared to those tabulated for airplane A (see Table 4.5). Very little difference between the eigenvalues is seen in the longitudinal modes. Slightly greater differences are noticed when comparing the lateral modes, but these differences are still fairly small. Another very good method

TABLE 4.5: EIGENVALUE COMPARISON FOR CESSNA 172 TEST CASE

Mode	Numerical	Independent
Longitudinal		
Short Period	$-4.130 \pm 4.3895j$	$-4.130 \pm 4.390j$
Phugoid	$-0.0209 \pm 0.1794j$	$-0.02092 \pm 0.1797j$
Lateral-Directional		
Dutch Roll	$-0.6947 \pm 3.3080j$	$-0.6858 \pm 3.306j$
Spiral	-12.4309	-12.43
Roll Response	-0.0109	-0.01095

for comparing the results of the numerical linearization with Roskam's tabulated data is to form plant and control matrices for the test aircraft, using the linear algebraic method taught in AE 3340 [Ref. Sch 92]. The resulting A and B matrices were then compared by applying step elevator, rudder, and aileron inputs and plotting the results. The results for the step elevator input are given in Figure 4.13. These plots show very little difference in the vertical velocity, w. Differing amplitudes are shown for the horizontal velocity, but since the non-dimensional velocity, u/V_T, from the

dimensional derivatives was scaled to be equivalent to the state, u, computed in the numerical linearization, these magnitude errors are not indicative of a poor model, only a slight difference in the computed damping is shown. The results from the analytic model were scaled up by the nominal airspeed, V_T to compare with the numerically linearized results. This would have the effect of magnifying any differences between the analytic model and the numerical model. The natural frequency of both the analytic and numerical results are quite similar.

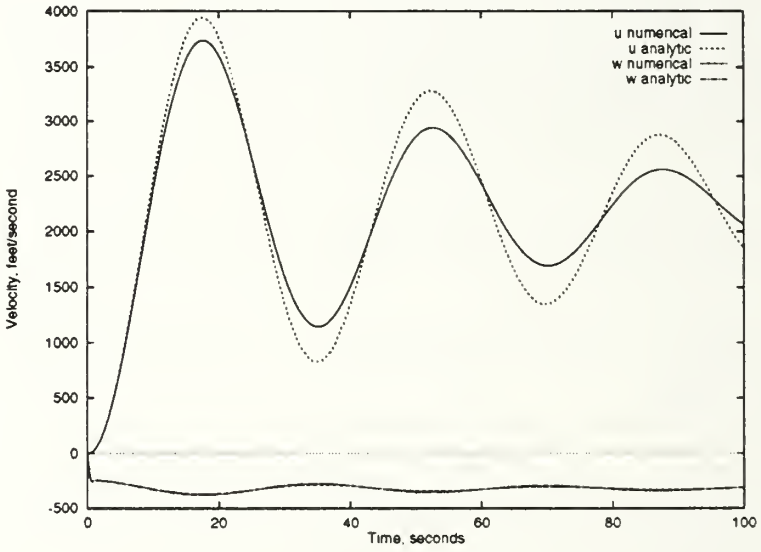

Figure 4.13: Comparison of Longitudinal Responses to Step Elevator Input

Lateral responses to step rudder and step aileron inputs are shown in Figures 4.14 and 4.15. Very little difference between the models is visible in these plots. The errors are shown in Figures 4.16 – 4.18.

It can be concluded that the results from the numerical linearization are quite close and furthermore that the equations used in the model are indeed correct. Moreover, the linearized results presented for both the AROD and Bluebird aircraft should be considered accurate and are suitable for the Guidance, Control, and Navigation system designs that will follow.

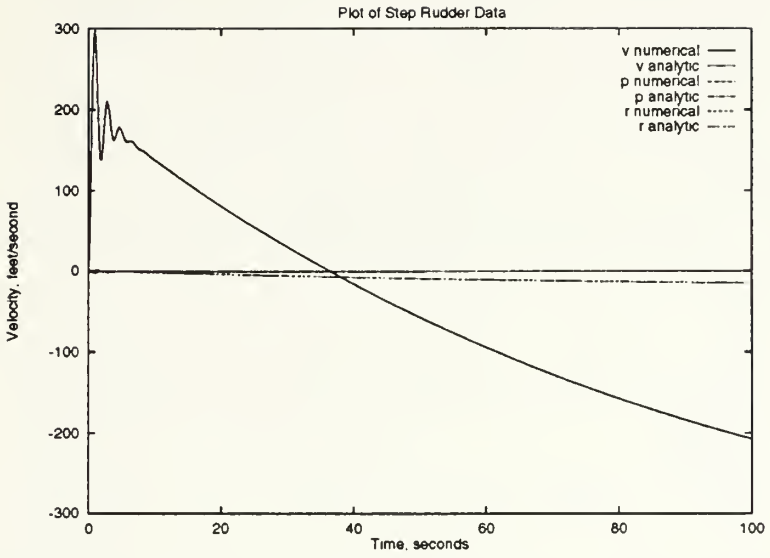

Figure 4.14: Comparison of Lateral Responses to Step Rudder Input

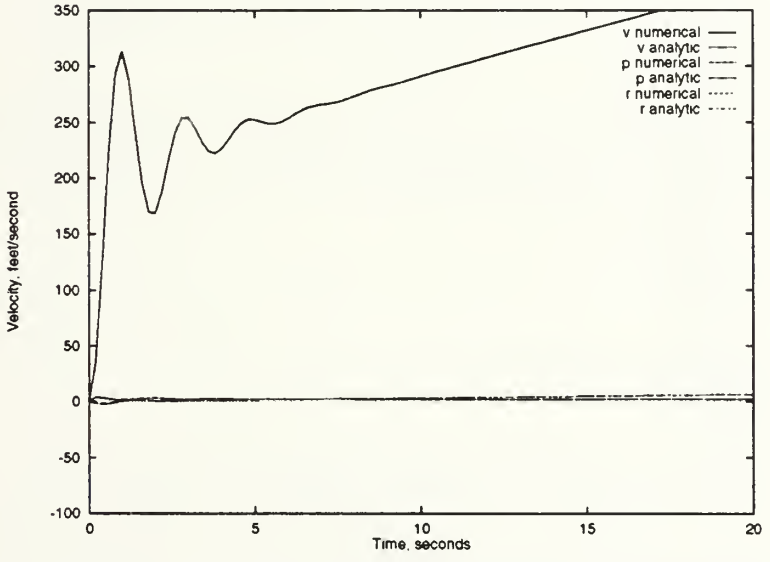

Figure 4.15: Comparison of Lateral Responses to Step Aileron Input

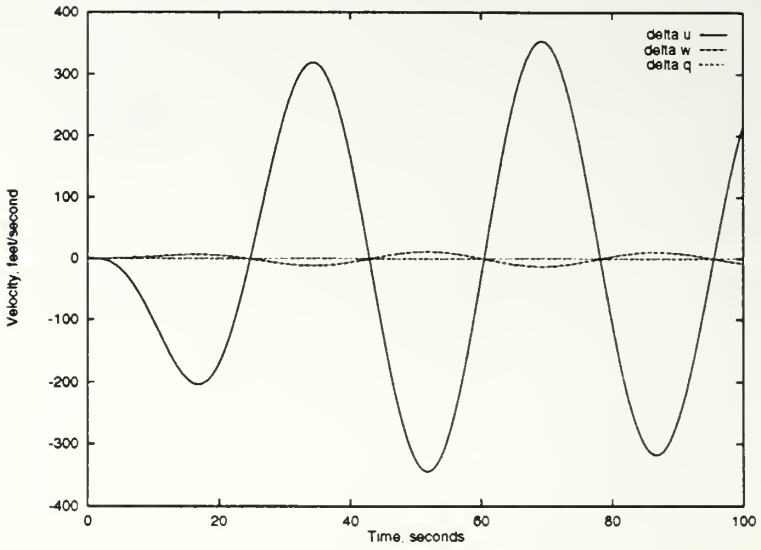

Figure 4.16: Difference in Analytic and Numerical Results, Step Elevator Input

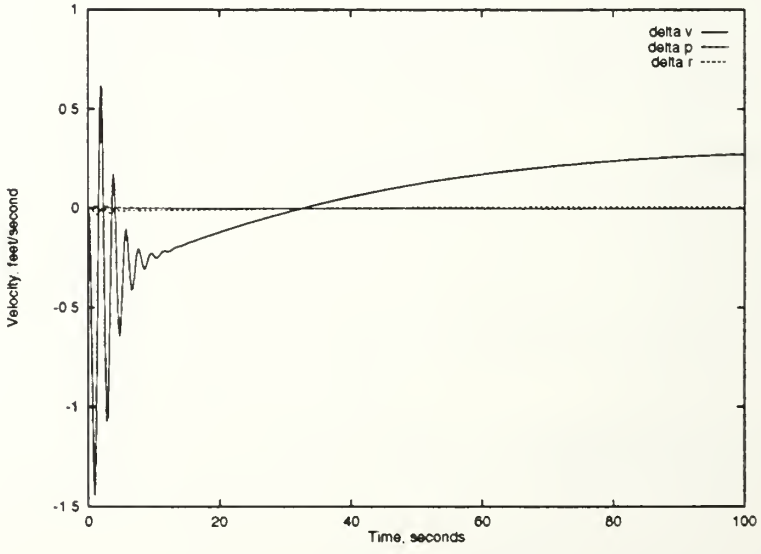

Figure 4.17: Difference in Analytic and Numerical Results, Step Rudder Input

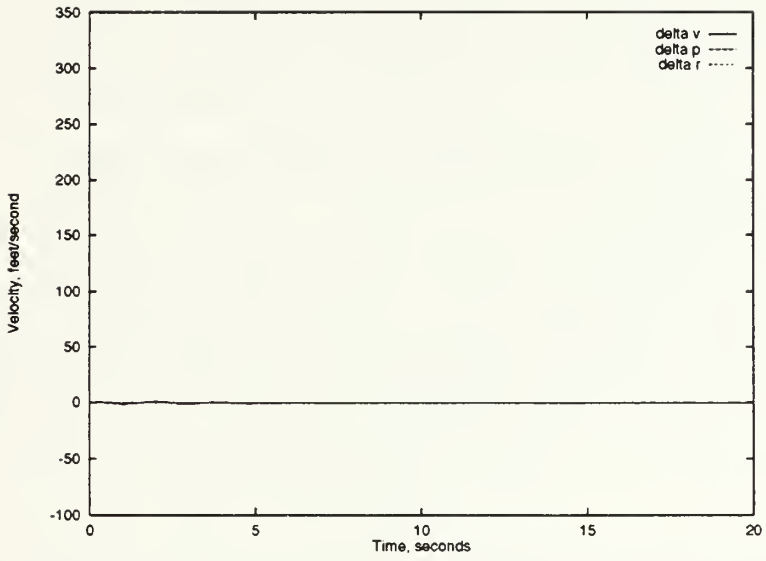

Figure 4.18: Difference in Analytic and Numerical Results, Step Aileron Input

V. SENSOR AND ACTUATOR MODELING

The AROD engine and actuators were modeled as a second order transfer functions, based on data collected by Sandia Labs [Ref. Wh 87]. Sensors were also modeled as second order transfer function based on data supplied by Watson Industries [Ref. WAT 93].

The complete non-linear model for an aircraft should include models of the sensors on board for measurement of acceleration, angular rates, pitch and roll angles, and headings. The inertial device chosen for the AROD project was the Watson Industries IMU-600D inertial measuring unit. This device contains a triaxial accelerometer, a triaxial rate sensor, two liquid pendulous devices (for bank and pitch angle), and a magnetic heading indicator. The characteristics of these devices must be accurately modeled, since it is the sensor output that drives the control system. In the rest of this section, the accelerometers, rate gyros, and inclinometers will be modeled, as well as the sources of error inherent in the design of the sensor devices.

A. ACCELEROMETER MODELING

The term accelerometer is not entirely accurate, since the device does not measure true acceleration, but rather the difference between acceleration and gravity [Ref. Bro 64]. This effect is referred to as the *Einstein Uncertainty Principle*, and is represented in equation form as

$$f = g - a \tag{5.1}$$

where g and a are the specific forces of gravity and acceleration of the aircraft and are measured positive downwards. The accelerometer model relevant to this equation is shown in Figure 5.1. The tri-axial accelerometer of the IMU-600D can be modeled

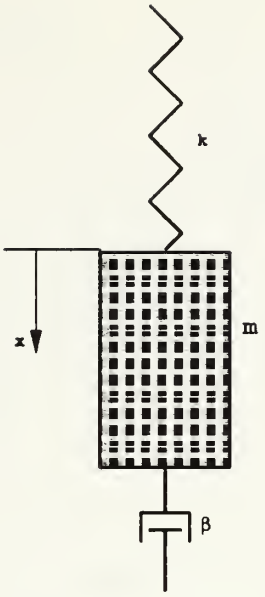

Figure 5.1: Typical Accelerometer Model

as three simple single–axis accelerometers, as has been established through conversations with the manufacturer [Ref. WAT 93]. A schematic representation basic device pictured in Figure 5.1 is modeled by an ordinary differential equation [Ref. Sil 91]

$$\ddot{x} + \frac{\beta}{m}\dot{x} + \frac{k}{m}x = -\ddot{y} \tag{5.2}$$

where x denotes the displacement of the mass from its equilibrium position and $\ddot{y} = g - a$ is the projection along the case axis of the vector sum of gravity and acceleration. The terms, β, k, and m represent the damping, spring coefficient, and mass, respectively, of the device. The accelerometer described in Equation 5.2 can be modeled as a second order low pass filter, but the actual accelerometer has a flat response up to 1000Hz, so it was not modeled. A third order Chebychev anti–aliasing filter with a cut–off frequency of 20 Hz was added to the accelerometers. This filter is the device that was modeled. The Chebychev filter gives the advantage of a flat passband, and a very sharp drop off at the cut–off frequency. The equation used to

describe a Chebychev filter [Ref. St 88] is

$$|H_{LP}(j\omega)|^2 = \frac{1}{1 + \epsilon^2 C_N^2(\omega)}, \tag{5.3}$$

where C_N is the N_{th} order Chebychev polynomial, ϵ is the parameter thats sets the ripple in the passband, and $|H_{LP}(j\omega)|^2$ is the magnitude of the filter. It was not necessary to compute the filter analytically, as SIMULINK provides a block function which performs the required steps based on the passband ripple of 0.1db and the desired cutoff frequency of 20 Hz. The block diagram for the accelerometer model is shown in Figure 5.2. Included in this diagram are the blocks containing the error modeling, as well as the blocks used to correct for the Einstein uncertainty. Figure 5.3 shows the linear synthesis model used to generate the accelerations to drive the sensor models. The synthesis model was derived from the Bluebird model discussed in Chapter IV.

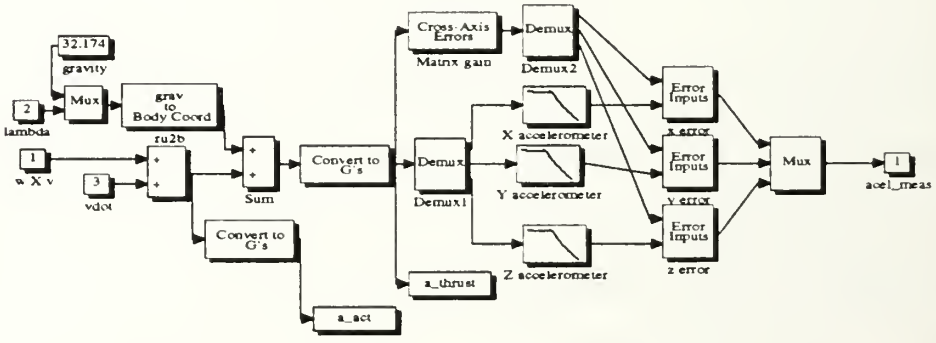

Figure 5.2: Accelerometer Modeling

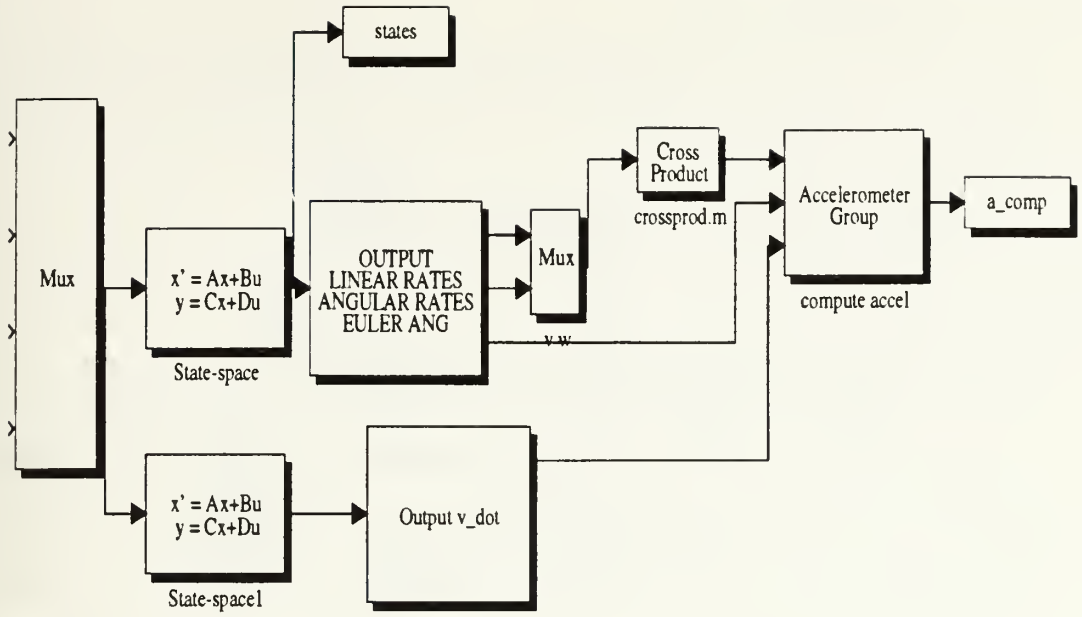

Figure 5.3: Synthesis Model for Accelerometers

1. Error Model

No matter how well the sensor device is constructed, any accelerometer is subject to certain errors in the linear acceleration measurements. These errors can occur for several reasons; some as mechanical and others are due to the physical placement of the accelerometer on the aircraft. The mechanical errors accounted for in the IMU–600D tri–axial accelerometers are

- Acceleration Bias

  – average readings for $+1g$ and $-1g$ loads

- Acceleration Scale Factor error

  –average difference between readings from $+1g$ and $-1g$ loads

- Cross Axis Sensitivity errors

  –measurements due to the misalignment of an accelerometer with the appropriate axis

  –measurements of off–axis accelerations are measured

- Acceleration Noise Floor

  –threshold below which measurements of acceleration can not be made

Errors in the measured accelerations can also occur if the accelerometers are not located at the c.g. of the aircraft, since arbitrarily located points on a body will experience additional accelerations due the the angular momentum of the body. A mathematical correction for the error will be presented later, and will describe the angular and linear accelerations of an arbitrary point on a rigid body. However, the correction will not be applied to the models here, since the correction is expected to have a very small effect on the sensor measurements due to the small displacements away from the aircraft c.g.

Error terms are quantified in terms of full–scale measurements. The mechanical errors are tabulated in Table 5.1 along with other important characteristics. Figure 5.4 shows the block diagram with error inputs applied to the accelerations

TABLE 5.1: ACCELEROMETER CHARACTERISTICS

Acceleration Range	$\pm 2g's$
Acceleration Bandwidth	20 Hz
Acceleration Bias	0.2% of Full Scale
Acceleration Scale Factor	0.2% of Full Scale
Acceleration Noise Floor	0.0005 $g's$
Cross Axis Sensitivity	0.5% of Full Scale

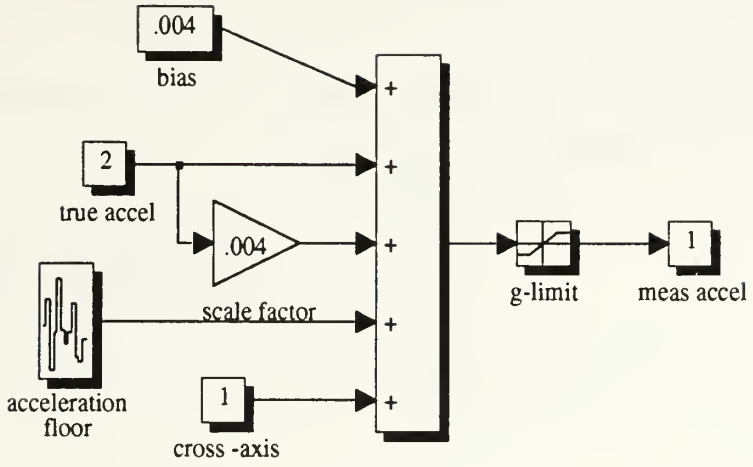

Figure 5.4: Error Model for Accelerometers

measured by the accelerometers. The output from the error computations in Figure 5.4 is the measured acceleration from the accelerometer output to the control system. The cross axis terms are determined through a matrix

$$\ddot{\tilde{x}} = \begin{bmatrix} 0 & e_y & e_z \\ e_x & 0 & e_z \\ e_x & e_y & 0 \end{bmatrix} \ddot{x} \tag{5.4}$$

where e_i is the cross axis error term and $\ddot{\tilde{x}}$ is the error in the acceleration due to the cross axis sensitivity.

2. Results and Validation

Accelerations were measured for step aileron, elevator, and rudder inputs, and the results then compared to the actual accelerations computed for the equations of motion for the aircraft. A linear synthesis model was used for the initial testing,

while the results from the non-linear model were used for validation of the accelerometer model. Figures 5.5, 5.6, and 5.7 show comparisons between measured and actual accelerations generated from simulations of the non-linear model. The accelerations computed for the longitudinal and lateral cases were in close agreement with the accelerations computed by the non-linear model.

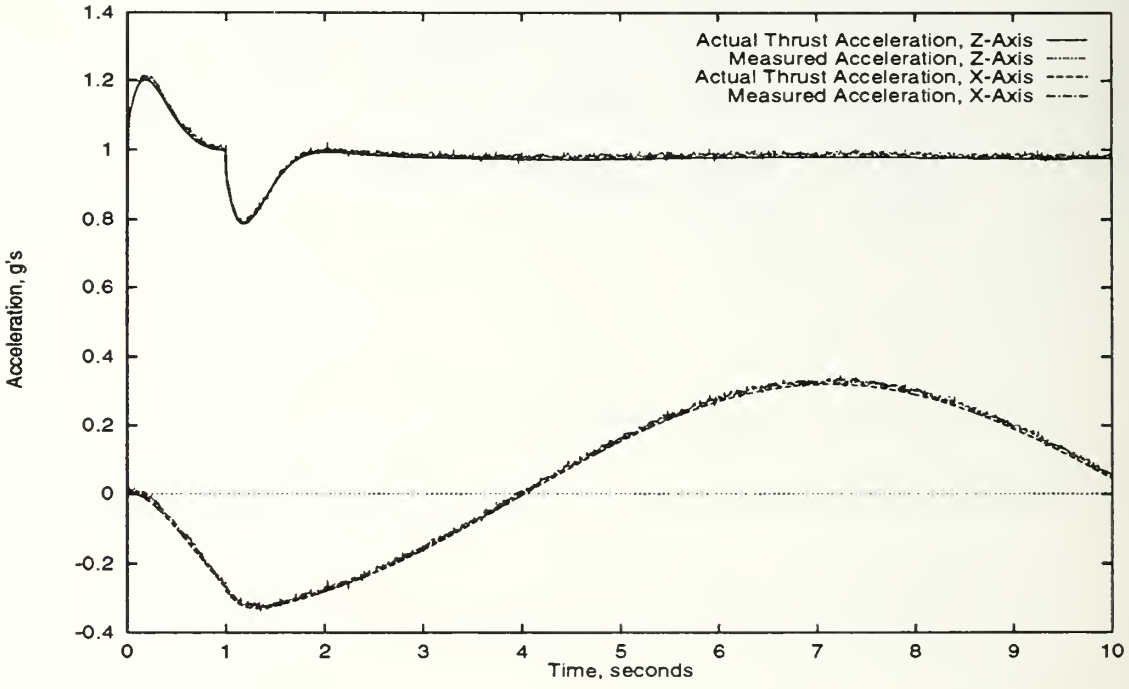

Figure 5.5: Measured and True Acceleration From a Step Elevator Input

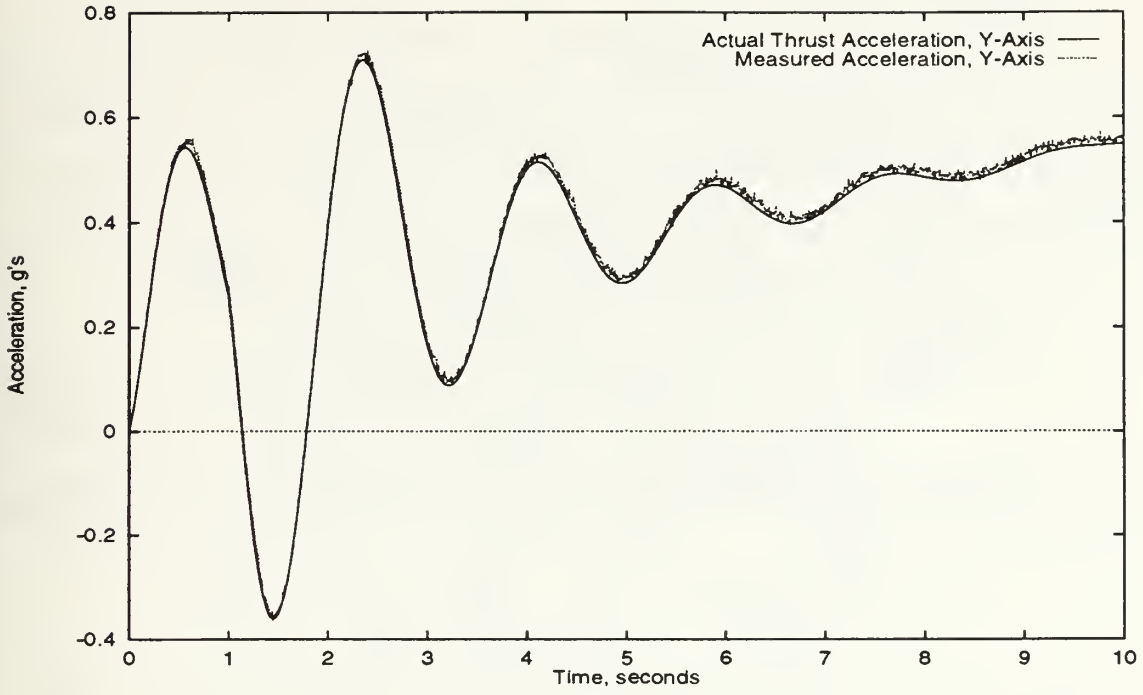

Figure 5.6: Measured and True Acceleration From a Step Aileron Input

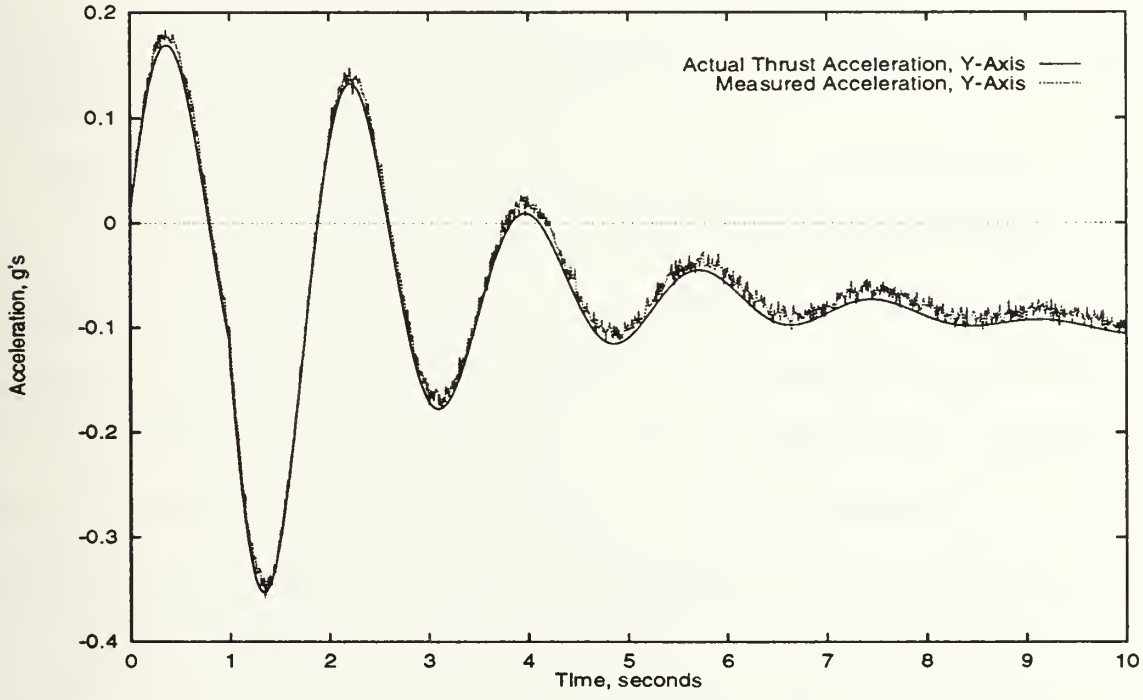

Figure 5.7: Measured and True Acceleration From a Step Rudder Input

B. RATE GYRO MODELING

The rate gyros consist of a rotating disk, mounted in a gimbal mechanism. Though both single and two degree of freedom gyros are common, the tri–axial angle rate gyro supplied in the IMU–600D is modeled as three single–degree–of–freedom rate gyros, each measuring the angular rate along a particular axis. The dynamics of a gyro can be modeled [Ref. Sil 91] using Euler's law

$$^{B}N_{G} = ^{B}\dot{L}_{G}$$

wher $\{B\}$ and $\{G\}$ are the body and gyro coordinate systems, respectively. This can be expanded to

$$^{B}N_{G} = ^{B}\omega_{G} \times ^{B}L_{G} + {}^{B}_{G}R\frac{d}{dt}{}^{G}L_{G}$$

where $^{G}L_{G} = I_{G}{}^{B}\omega_{G}$ and the time derivative, $\frac{d}{dt}{}^{G}L_{G}$ is zero when the wheel rotates at a constant speed. Thus, except for the period when the wheel is coming up to speed, the equation for gyroscopic motion in a rate gyro can be written as

$$^{B}N_{G} = ^{B}\omega_{G} \times ^{B}L_{G} \tag{5.5}$$

where $^{B}N_{G}$ is the torque vector acting on the gyro element and $^{B}\omega_{G}$ is the angular rate of the gyro frame. Transfer functions can be developed for the torque input to the input axis as shown in Figure 5.8 and the rate output of the output axis. This derivation was not developed since data required for the gyro disk inertia, I_G and the speed of rotation, $^{B}\omega_{G}$, among other terms, is not available from the manufacturer. The rate gyros were modeled as second order transfer functions, with ω_n = 50 Hz. A third order Chebychev filter with a cut–off frequency of 20 Hz was added to the gyro model as an anti–aliasing filter. This Chebyshev filter was identical to the ones described for the accelerometers. The block diagram of the rate gyro model is shown in Figure 5.9. The linear synthesis model is shown in Figure 5.10 and was used to

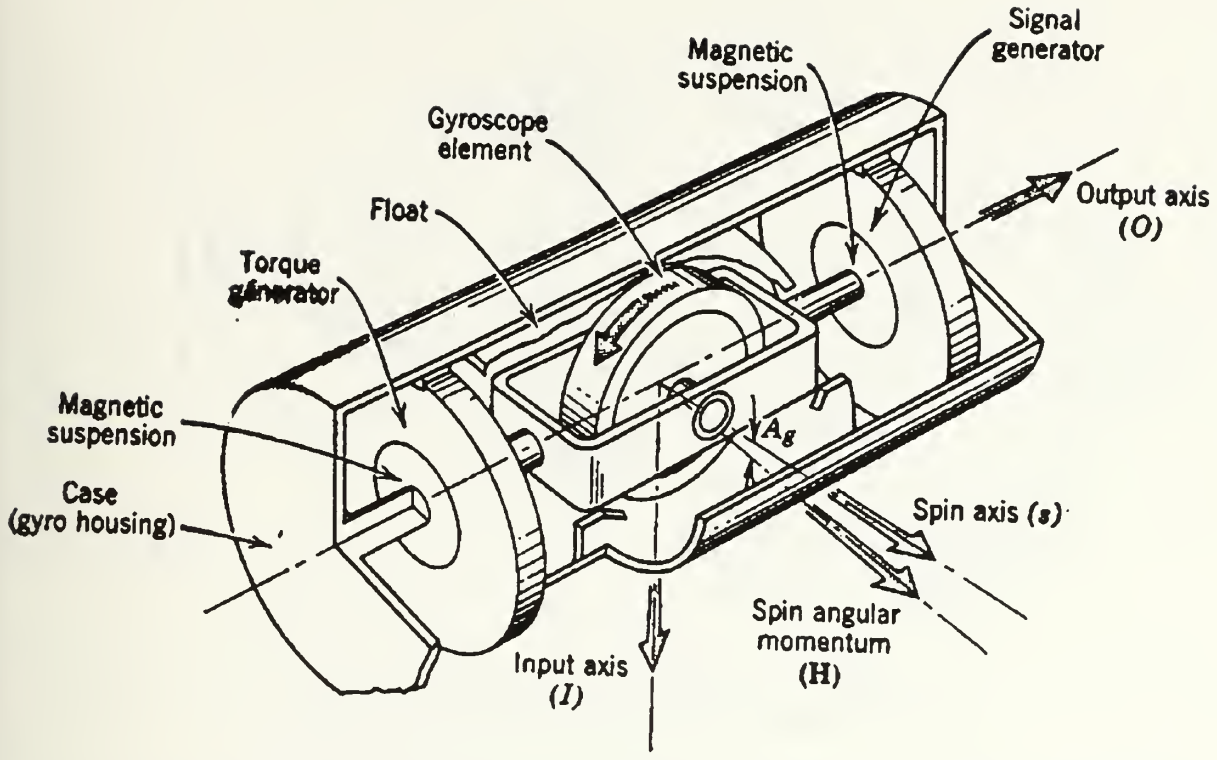

Figure 5.8: Functional Diagram of a Rate Gyro [Ref. Bro 64]

test the gyro models. In Figure 5.9 the gyro elements are shown for each axis and blocks for the error calculations are shown as well.

1. Error Modeling

Error terms are also present in the rate gyros and are due to either physical location on the aircraft, or mechanical errors, as in the accelerometers. Errors due to physical placement away from the c.g. can be corrected by using the equations derived for the linear and angular acceleration of an arbitrary point on the aircraft, shown later in this chapter. Mechanical errors are defined in the same manner as was done for accelerometers in Section A, and are listed in Table 5.2 along with other

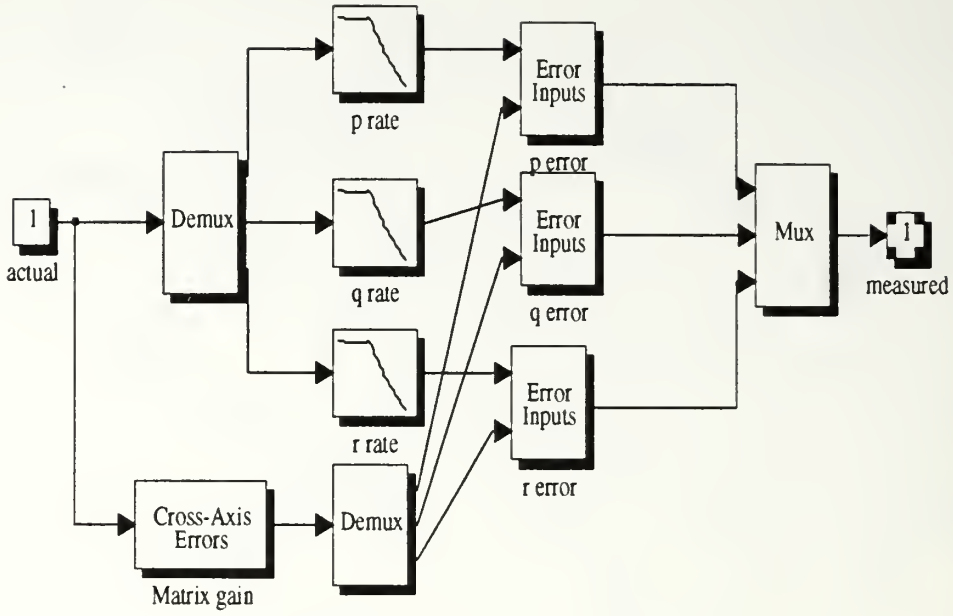

Figure 5.9: Rate Gyro Model

TABLE 5.2: GYRO CHARACTERISTICS

Rotational Rate Range	$\pm 114.6 deg/sec$
Rotational Rate Bandwidth	20 Hz
Rotational Rate Bias	2.0% of Full Scale
Rotational Rate Scale Factor	0.5% of Full Scale
Rotational Rate Noise Floor	0.05% of Full Scale
Cross Axis Sensitivity	0.5% of Full Scale

important characteristics. The cross axis error is modeled in the same way as for the accelerometers by the use of the same matrix for computing errors in the angular rates.

2. Results and Validation

Angular rates were measured for step aileron, elevator, and rudder inputs and compared to the actual angular rates computed from the equations of motion. First a linearized synthesis model was used in the testing stages; then the sensors

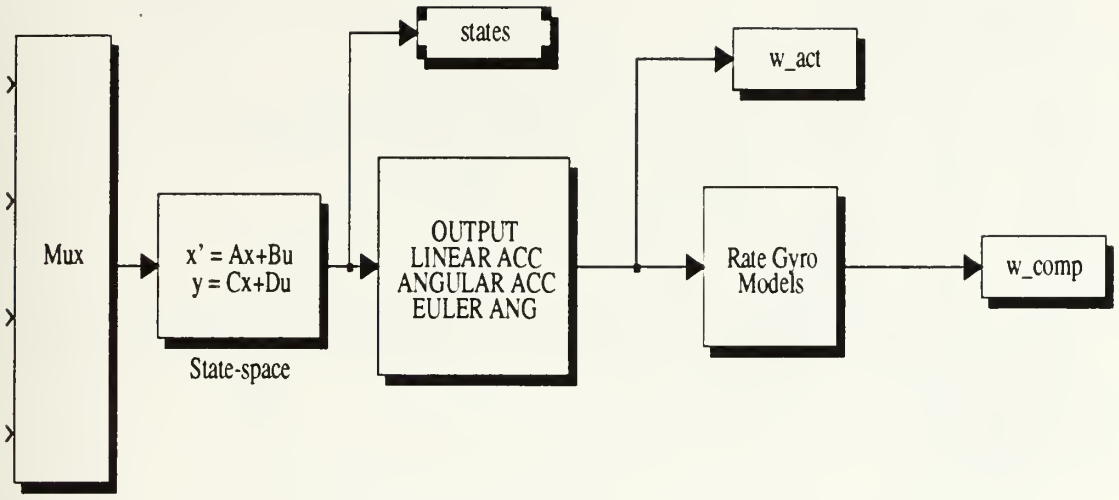

Figure 5.10: Rate Gyro Synthesis Model

were integrated with the non-linear aircraft simulation. Comparisons of actual and measured acceleration are given in Figures 5.11, 5.12, and 5.13. These figures show the accuracy of the angular rate sensors in measuring the angular rates computed with the non-linear equation of motion model.

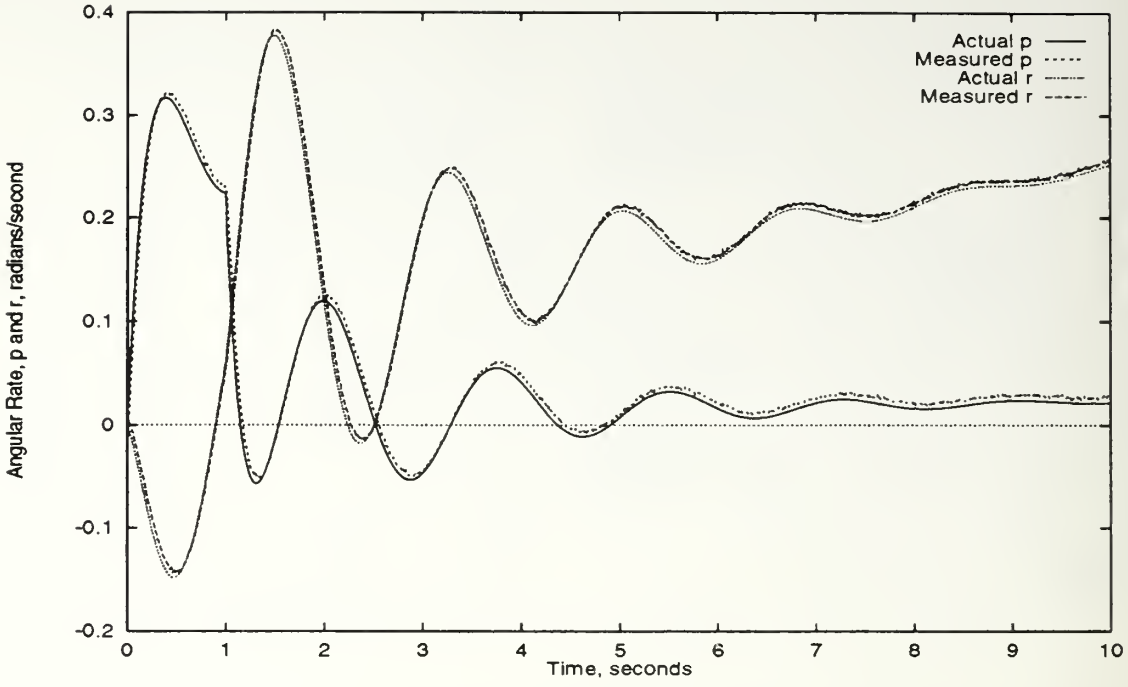

Figure 5.11: Measured and True Angular Rates From a Step Aileron Input

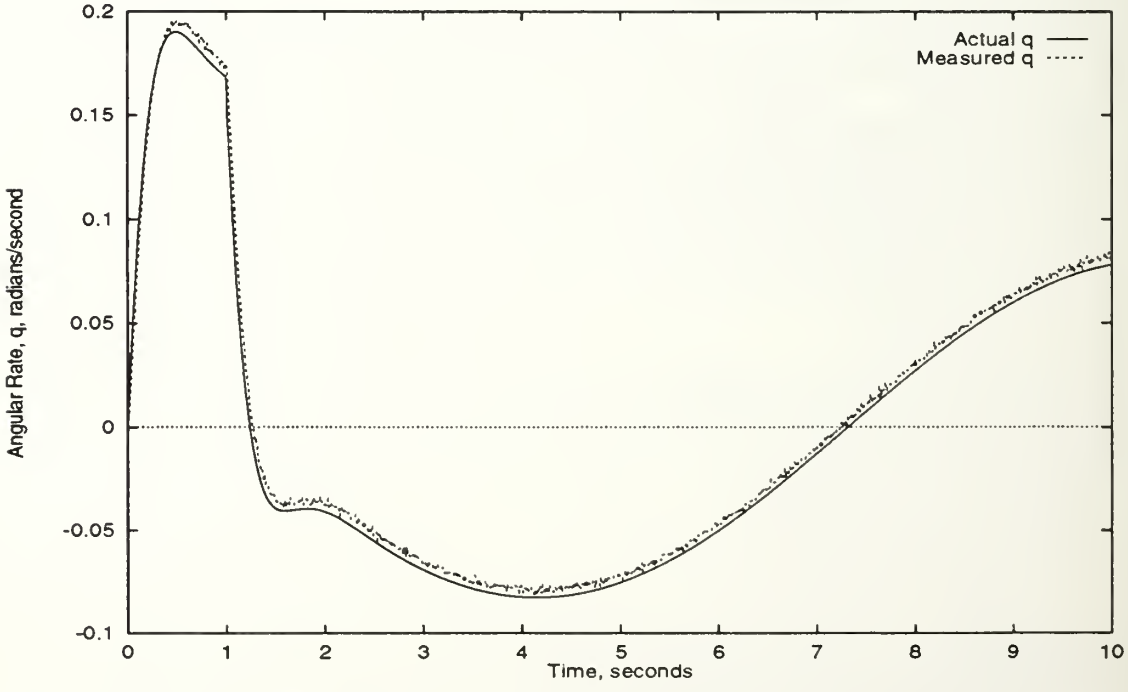

Figure 5.12: Measured and True Angular Rates From a Step Elevator Input

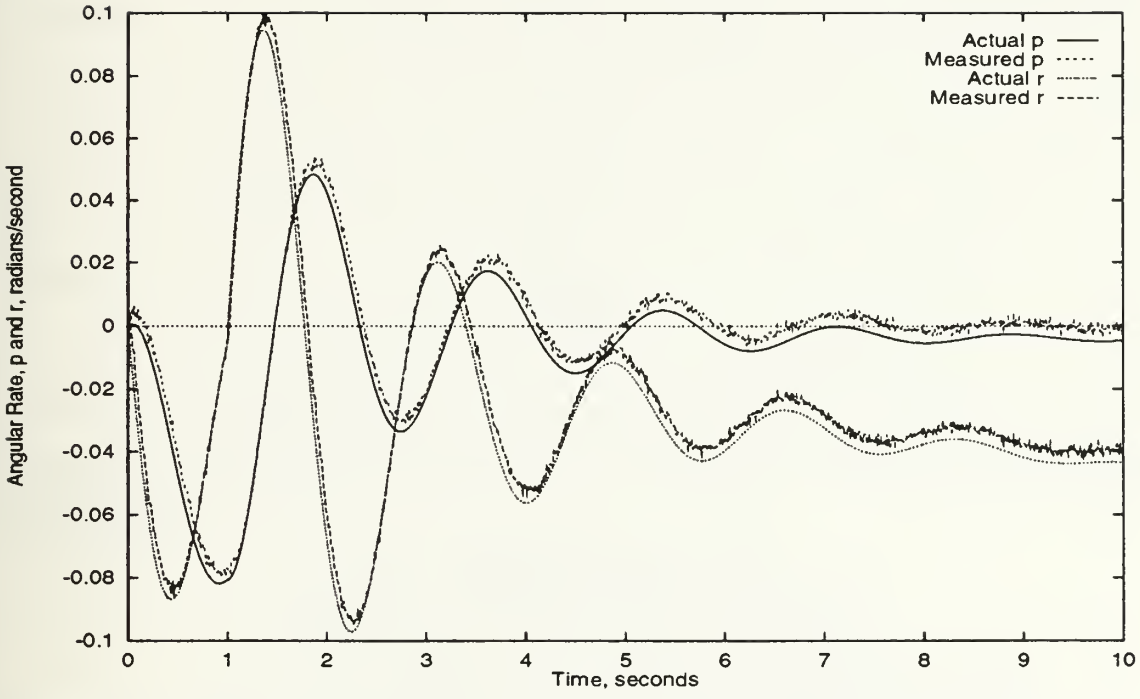

Figure 5.13: Measured and True Angular Rates From a Step Rudder Input

C. PITCH, ROLL, AND HEADING SENSOR MODELING

The last group of sensors to be modeled are those which measure the pitch, roll, and heading angles. The pitch and roll angle measurements are made with liquid pendulous devices. These are devices that have an electrolyte contained in a vial, and by measuring the capacitance changes of the vial as the electrolyte moves in response to aircraft angle changes, the pitch and roll angles can be determined. The heading sensor is a magnetic heading sensor.

The primary concern with angular measurements is that accurate readings are obtained, even when the aircraft is experiencing linear and angular accelerations. When these errors cannot be corrected, the control system must be able to compensate for the errors. An inclinometer is typically modeled as a pendulum, shown in Figure 5.14 attached to a block, which can be considered to represent an aircraft. The equations describing the motion of the pendulum are derived in detail later in this section. The transfer function for the pendulum inclinometer can be represented as a second order transfer function with $\omega_n = 0.8$ Hz and $\zeta = 0.5$. [Ref. WAT 93]

$$\frac{\hat{\Theta}}{\Theta} = \frac{5.03^2}{s^2 + 50.3s + 5.03^2}.$$

1. Error Modeling

The inclinometers are subject to several sources of errors, from both linear and angular accelerations, and from mechanical imperfections. Mechanical errors are listed in Table 5.3, and modeled as shown in Figure 5.15. Errors due to angular velocity can be compensated for with a complementary filter. Selecting the time constants appropriately will allow direct measurements of the angle from the inclinometer at low frequencies, and from integrated angular rates at higher frequencies where the inclinometer is inaccurate.

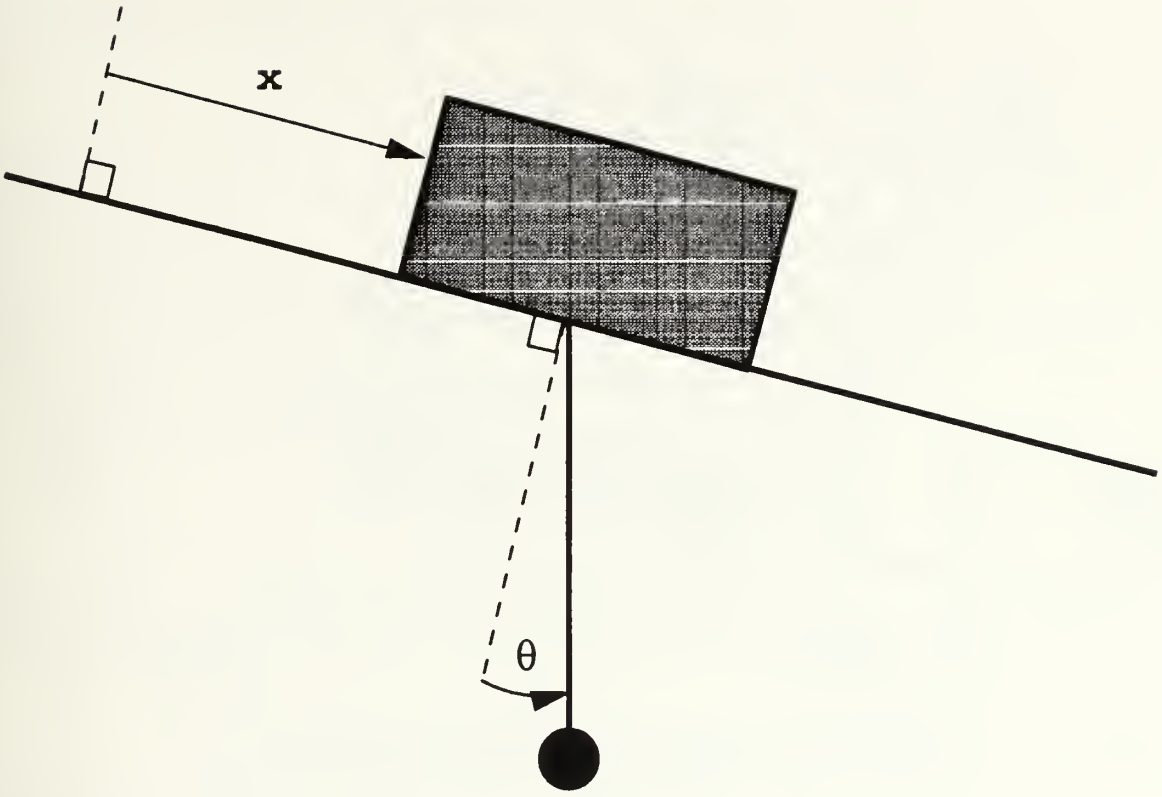

Figure 5.14: Simple Pendulum Inclinometer

TABLE 5.3: INCLINOMETER AND HEADING SENSOR CHARACTERISTICS

Pitch and Roll Range	± 50 *deg*
Pitch and Roll Bandwidth	1/2 Hz
Pitch and Roll Accuracy	0.2 *deg*
Heading Range	± 180 *deg*
Heading Accuracy	3.0 *deg*
Heading Repeatability	0.5 *deg*
Heading Linearity	0.5%

It was determined that the effects of linear acceleration were also important to model. In order to model these effects it was necessary to derive a transfer function from the aircraft's linear acceleration to the angle caused by that acceleration. The starting point was to model the inclinometer as shown in Figure 5.14. The derivation

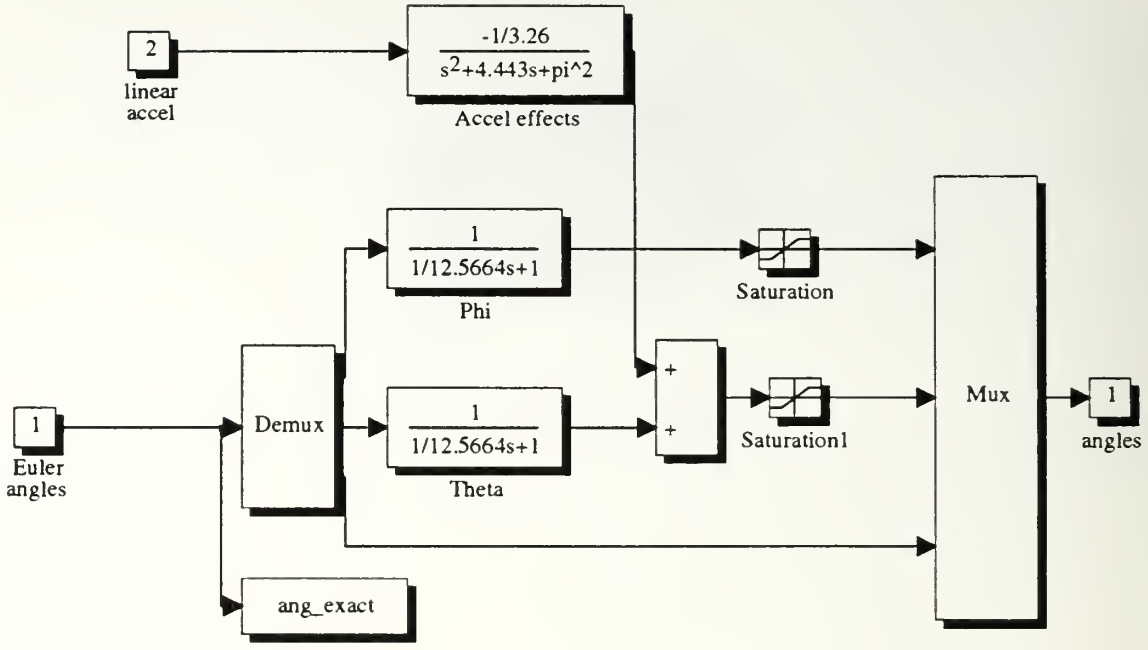

Figure 5.15: Inclinometer Error Modeling

results in two coupled equations, the first of which will be ignored, since the equations of motion for the aircraft are known and the motion of the inclinometer should have little effect on the aircraft motion. The second equation is for the motion of the pendulum as influenced by the aircraft's acceleration and is what was used for the linear acceleration errors.

First it was necessary to define the coordinates used to describe the inclinometer system as $q = (x, \theta)$ (see Figure 5.14). Next, since Lagrangian methods were used for the derivation, the kinetic energy, T, of the system was defined as

$$\begin{aligned} T &= 1/2M\dot{x}^2 + 1/2m(\dot{x} + l\dot{\theta}\cos\theta)^2 - 1/2m(l\dot{\theta}\sin\theta)^2 \\ &= 1/2M\dot{x}^2 + 1/2m\dot{P}_x^2 + 1/2m\dot{P}_y^2, \end{aligned} \tag{5.6}$$

where P_x and P_y represent the position of the pendulum in Cartesian coordinates.

The potential energy, V, of the pendulum can be written as

$$V = V_m = mgl(1 - \cos\theta), \tag{5.7}$$

where M, m, and l are the mass of the aircraft, mass of the pendulum, and distance from the aircraft c.g. to the pendulous mass. The terms,P_i and θ, represent the position vector to the pendulous mass and the angle made by the pendulum with the vertical plane. The governing equation for Lagrangian dynamics is

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i \tag{5.8}$$

where $\mathcal{L}$ is the Lagrangian operator, $\mathcal{L} = T - V$, and Q_i represents the non-conservative forces acting on the body. After some rearranging,

$$\mathcal{L} = 1/2(M + m)\dot{x}^2 + ml\dot{x}\dot{\theta}\cos\theta + 1/2ml^2\dot{\theta}^2 - mgl(1 - \cos\theta) \tag{5.9}$$

The result in Equation 5.9 can be substituted into Equation 5.8 resulting in

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \dot{x}} &= (M + m)\dot{x} + ml\dot{\theta}\cos\theta \\
\frac{\partial \mathcal{L}}{\partial \dot{\theta}} &= ml\dot{x}\cos\theta + l^2\dot{\theta} \\
\frac{\partial \mathcal{L}}{\partial x} &= 0 \\
\frac{\partial \mathcal{L}}{\partial \theta} &= -ml\dot{x}\dot{\theta}\,\sin\theta - mgl\sin\theta \\
\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} &= (M + m)\ddot{x} + ml\cos\theta\,\ddot{\theta} - ml\sin\theta\,\dot{\theta}^2 \\
\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}} &= ml\cos\theta\ddot{x} - ml\sin\theta\,\dot{\theta} + ml^2\ddot{\theta}.
\end{aligned}$$

When these partial derivatives are substituted into Equation 5.8, the resulting equations of motion for the pendulum are

$$\begin{aligned}
(M + m)\ddot{x} + ml\cos\theta\,\ddot{\theta} - ml\sin\theta\,\dot{\theta}^2 &= Q_1 \\
ml\cos\theta\ddot{x} + ml^2\ddot{\theta} + mgl\sin\theta &= Q_2,
\end{aligned}$$

where Q_1 and Q_2 are terms representing damping in the system. The term Q_1 represents damping on the aircraft for $Q_1 = -\beta_x \dot{x}$ and the term Q_2 represents the viscous damping of the pendulous mass, $Q_2 = -\beta_r \dot{\theta}$. The equations can be linearized, where

$$(M+m)\ddot{x} + ml\ddot{\theta} + \beta_x \dot{x} = 0 \tag{5.10}$$

$$ml\ddot{x} + ml^2\ddot{\theta} + \beta_r \dot{\theta} + mgl\theta = 0. \tag{5.11}$$

Now it is apparent that Equation 5.10 is completely determined by the aircraft equations of motion that have already been modeled. A Laplace transform of Equation 5.11 can be performed to find the desired transfer function

$$\frac{\Theta}{\ddot{x}} = \frac{-1/l}{s^2 + \frac{\beta_r}{ml^2}s + g/l}, \tag{5.12}$$

where the similarity to the standard second order transfer function is noted, making the term $g/l = \omega_n^2$. The term $-1/l$ in the numerator can be solved for by substituting $g = 32.174\ f/s^2$ and $\omega_n = 5.027\ rad/sec$, or $0.8Hz$, with the result $l = 1.27\ f$. This result is then substituted into the block diagram for the inclinometer models shown in Figure 5.15. Responses for step inputs are shown in Figures 5.16, 5.17, and 5.18. In the lateral cases, there is very little difference between the measured and actual angles. The longitudinal case is quite different. In Figure 5.16, there is a considerable difference between the actual pitch angle, Θ, and the pitch angle measured with the inclinometer model. This difference the linear acceleration of the aircraft as a step elevator input is applied, and must be compensated for before a reliable control system can be developed.

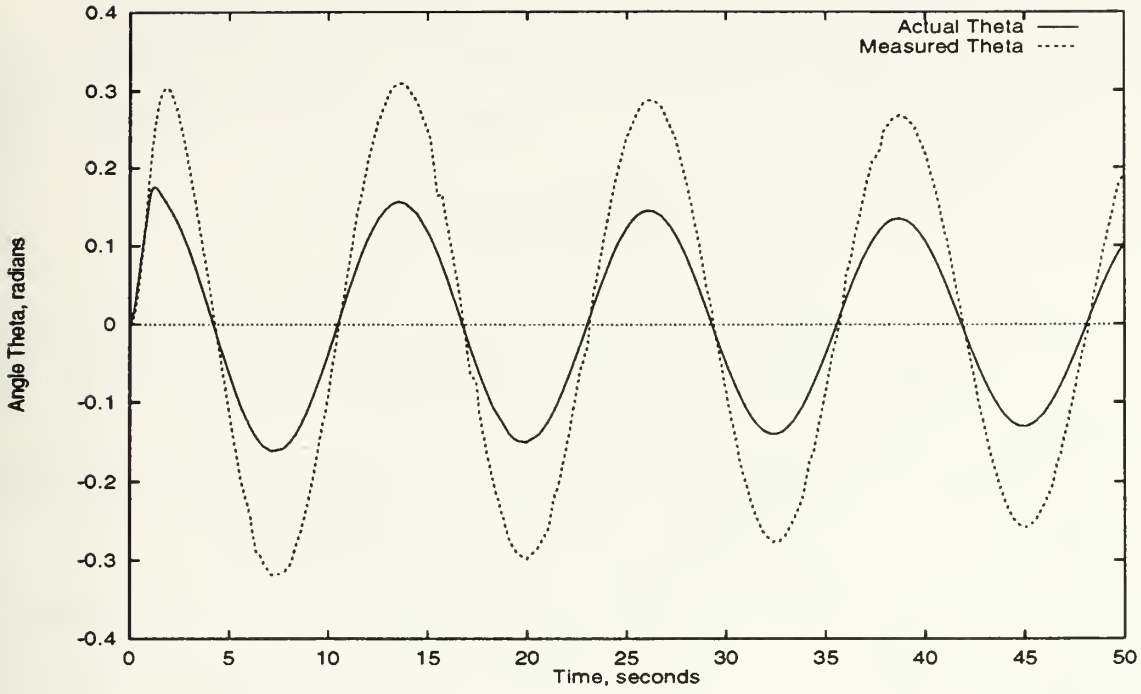

Figure 5.16: Response to a Step Elevator Input

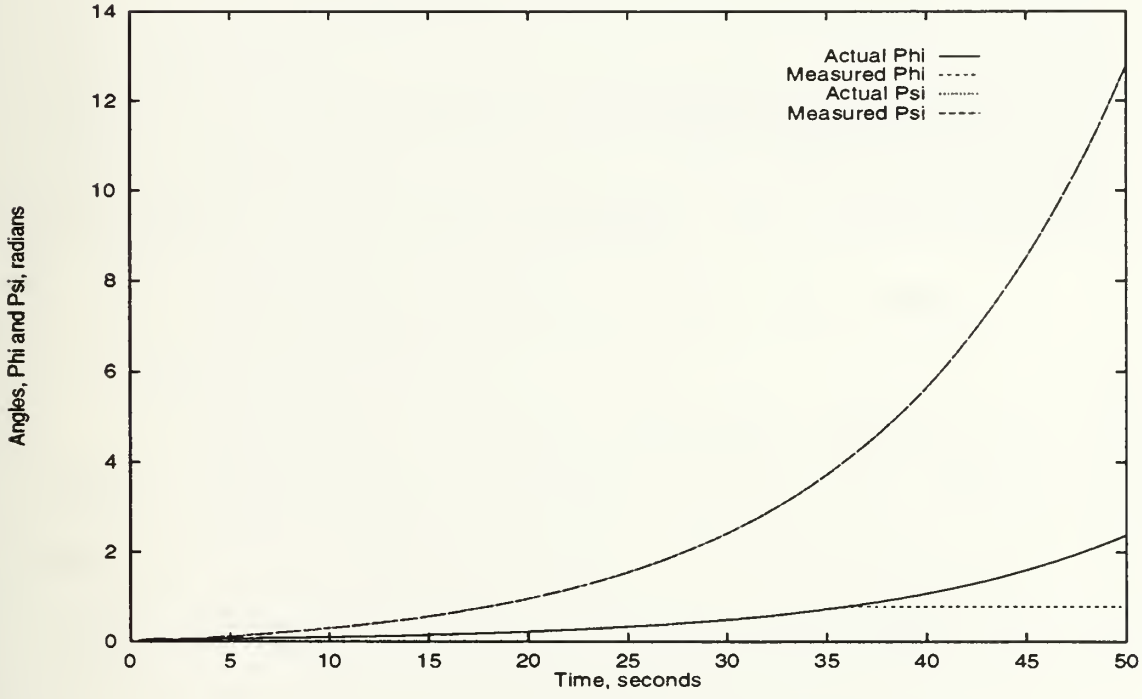

Figure 5.17: Response to a Step Rudder Input

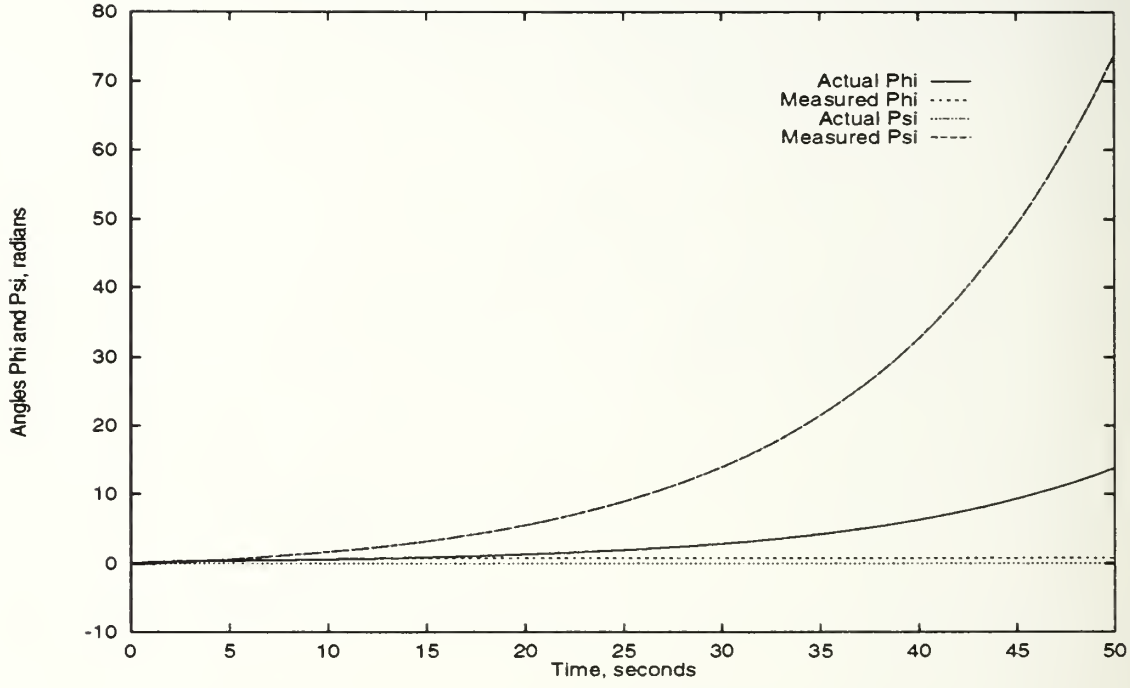

Figure 5.18: Response to Step Aileron Input

The error in inclinometer readings due to linear acceleration has been determined experimentally by the manufacturer, and can be represented as a second order transfer function from $g's$ to Θ,

$$\frac{\Theta}{g} = K\frac{157.08^2}{s^2 + 157.08s + 157.08^2}, \tag{5.13}$$

where the gain, K, was determined from an input of 1.2 V at 10 mV/g, and the resulting output of 0.4 V at 60 mV/deg. Thus the gain, K, was 17.99.

D. MODELING OF AN ARBITRARY SENSOR PLACEMENT

The actual sensor placement in either the Bluebird or the AROD aircraft is not at the c.g., as was assumed in the previous sections. In order to model the actual linear and angular accelerations at the sensors, regardless of the position on the aircraft, new equations of motion must be derived. These equations can then be applied to the sensor inputs to obtain a proper output from the sensor. First the equations for linear acceleration will be derived using Newton's third law for conservation of momentum, then the equations for angular acceleration will be derived using Euler's law for conservation of angular momentum. These equations must be expressed in terms of $\{B\}$ since all the information measured by the sensors will be determined in the body coordinate system.

1. Linear Accelerations

In the derivation for equations of motion of an arbitrary point on the aircraft, the coordinate systems representing the inertial and body coordinate systems, similar to those shown in Figure 3.1 are used. Supplementing the derivation of equations of motion for an aircraft, the motion of the sensor location on the aircraft, given by P_Q is necessary. In the inertial coordinate system, $\{U\}$, the position of Q can be written as ${}^{U}P_Q = {}^{U}_{B}R\,{}^{B}P_Q + {}^{U}P_{BO}$. The velocity is first determined by applying

Coriolis' theorem from Equation 3.24 to obtain the first derivative. Here,

$$ {}^{U}V_Q \triangleq {}^{U}\dot{P}_Q = {}^{U}\dot{P}_{BO} + {}^{U}\frac{d}{dt}({}^{U}_{B}R\,{}^{B}P_Q), \tag{5.14} $$

where

$$ {}^{U}\frac{d}{dt}({}^{U}_{B}R\,{}^{B}P_Q) = {}^{B}\frac{d}{dt}({}^{U}_{B}R\,{}^{B}P_Q) + {}^{U}\omega_B \times ({}^{U}_{B}R\,{}^{B}P_Q). \tag{5.15} $$

The terms ${}^{U}\frac{d}{dt}$ and ${}^{B}\frac{d}{dt}$ refer to derivatives taken with respect to the inertial and body coordinate systems, respectively. The resulting expression for velocity is

$$ {}^{U}V_Q = {}^{U}V_{BO} + {}^{\omega}_{B} \times ({}^{U}_{B}R\,{}^{B}P_Q), \tag{5.16} $$

where ${}^{B}\frac{d}{dt}({}^{U}_{B}R\,{}^{B}P_Q) = 0$ since Q is a point fixed on the aircraft. Accelerations are derived by applying Equation 3.24 twice to Equation 5.16

$$ {}^{U}\dot{V}_Q = {}^{B}\frac{d}{dt}({}^{U}V_{BO} + {}^{\omega}_{B} \times ({}^{U}_{B}R\,{}^{B}P_Q)) + {}^{\omega}_{B} \times ({}^{U}V_{BO} + {}^{\omega}_{B} \times ({}^{U}_{B}R\,{}^{B}P_Q)). \tag{5.17} $$

The ${}^{B}\frac{d}{dt}(\cdot)$ term is differentiated, resulting in

$$ {}^{B}\frac{d}{dt}({}^{U}V_{BO} + {}^{\omega}_{B} \times ({}^{U}_{B}R\,{}^{B}P_Q)) = {}^{B}\frac{d}{dt}({}^{U}V_{BO}) + {}^{U}\dot{\omega}_B \times {}^{U}_{B}R\,{}^{B}P_Q + {}^{\omega}_{B} \times {}^{U}V_Q. \tag{5.18} $$

Substituting the results from Equation 5.18 into Equation 5.17, the result is

$$ {}^{U}\dot{V}_Q = {}^{B}\frac{d}{dt}({}^{U}V_{BO}) + 2\,{}^{U}\omega_B \times {}^{U}V_Q + {}^{U}\dot{\omega}_B \times {}^{U}_{B}R\,{}^{B}P_Q + {}^{\omega}_{B} \times ({}^{\omega}_{B} \times {}^{U}_{B}R\,{}^{B}P_Q). \tag{5.19} $$

The desired result in $\{B\}$ is obtained by simply premultiplying Equation 5.19 by ${}^{U}_{B}R$

$$ {}^{B}\dot{V}_Q = \frac{d}{dt}({}^{B}v_{BO}) + 2\,{}^{B}\omega_B \times {}^{B}V_Q + {}^{B}\dot{\omega}_B \times {}^{B}P_Q + {}^{B}\omega_B \times ({}^{B}\omega_B \times {}^{B}P_Q), \tag{5.20} $$

where the identity [Ref. Sp 89]

$$ {}^{B}_{U}R(\omega_B \times {}^{U}V_Q) = ({}^{B}_{U}R\omega_B) \times ({}^{B}_{U}R\,{}^{U}V_Q) $$

was used. This result can now be substituted into Equation 3.27 and the resulting expression can be solved for $\frac{d}{dt}({}^{B}v_{BO})$.

2. Angular Accelerations

No further work is needed to derive the expressions for angular velocity and acceleration at a arbitrary point, since for a rigid body, these quantities remain the same anywhere on the body.

$$\omega_{BO} = \omega_Q \; \forall Q \in \{B\}$$

$$\dot{\omega_{BO}} = \dot{\omega_Q} \; \forall Q \in \{B\},$$

since $\omega \times \omega = 0$.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Based on the data presented in this thesis, the following conclusions are made:

- High–fidelity models of both the AROD and Bluebird aircraft were implemented using SIMULINK software. Use of the block diagram structure of the model allows changes to be easily made and requires no programming ability, other than the use of MATLAB.

- These models accurately represent the sensors, actuators, and engines associated with the particular aircraft.

- Expected errors in the accelerometers and rate gyros are very small. Pitch errors from the inclinometers, due to linear accelerations, will be much greater.

- The models that were developed only represent one flight condition. The AROD was modeled only in a hover and the Bluebird was only modeled in a cruise condition. Further work on the control, guidance, and navigation systems will be concentrated in these flight regimes. The data files for a given flight condition are easily replaced with tables, when more test data are available.

B. RECOMMENDATIONS

Based on the conclusions presented above, and the problems encountered during this study, the following recommendations are made.

- The Department of Aeronautics and Astronautics should update the UNIX labs to include SIMULINK with MATLAB 4.0. This will allow increased instructional

use in flight dynamics, controls, and avionics courses. Additionally, a practical research tool would be available.

- The work in this thesis must be modified to include the aerodynamic characteristics of the Archytas aircraft. Achieving this step will require wind tunnel testing for the Archytas in all the expected phases of flight, especially the extremely non–linear transition phase from vertical to horizontal flight.

- Integration of the quaternion based rotation matrix should be completed in order to take advantage of the superior qualities of quaternions in computational models.

APPENDIX A: MATHEMATICAL PROPERTIES

In this appendix, some of the mathematical properties used in the text are described.

A. CROSS PRODUCT PROPERTIES

In this section, important properties of the cross product and cross product matrices are described The cross product between two vectors,

$$A = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \text{ and } B = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$$

is defined by

$$A \times B = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix}$$

Properties of the cross product are:

- $$A \times B = (A\times)B, \text{ where } A\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

  and is called a cross product matrix

- ${}^A_B R(V \times U) = ({}^A_B RV) \times ({}^A_B RU)$ if ${}^A_B R$ is a rotation matrix with V and U in the same coordinate system. That this matrix distributes across the cross product is obvious since rotation matrices preserve space geometry.

- ${}^A_B R(V \cdot U) = ({}^A_B RV) \cdot ({}^A_B RU)$ for the same reasons as above.

- $A \times (B \times C) = (A \cdot C)B - (A \cdot B)C$

- $A \times (B \times C) = B \times (A \times C) + C \times (B \times A)$
- $A \times B = -B \times A$
- $-A\times = A^T\times$

B. DERIVATIVES OF VECTORS

For any free vector, ${}^BV_Q$ (i.e. a velocity vector and any rotation matrix, ${}^A_BR$, the derivative of the velocity of Q computed in $\{B\}$ and expressed in $\{A\}$ and denoted as ${}^A({}^BV_Q)$ is given as [Ref. Sil 91]

$$\begin{aligned}\frac{d}{dt}({}^A({}^BV_Q)) &= \frac{d}{dt}({}^A_BR\ {}^BV_B) \\ &= {}^A_B\dot{R}\ {}^BV_Q + {}^A_BR\frac{d}{dt}({}^BV_Q) \\ &= ({}^A_B\dot{R}\ {}^B_AR){}^A_BR\ {}^BV_Q + {}^A_BR\frac{d}{dt}({}^BV_Q)\end{aligned}$$

since as shown in [Ref. Cra 86],

$$\frac{d}{dt}({}^A({}^BV_Q)) = {}^A\Omega_B \times {}^A({}^BV_Q) + {}^A_BR\frac{d}{dt}({}^BV_Q)$$

so then

$${}^A_B\dot{R}\ {}^B_AR = {}^A\Omega_B\times$$

The same process can be carried out for an angular velocity vector ${}^A\Omega_B$,

$$\begin{aligned}\frac{d}{dt}({}^A\Omega_B) &= \frac{d}{dt}({}^A_BR\ {}^B({}^A\Omega_B)) \\ &= {}^A\Omega_B \times {}^A\Omega_B + {}^A_BR\frac{d}{dt}({}^B({}^A\Omega_B)) \\ &= {}^A_BR\frac{d}{dt}({}^B({}^A\Omega_B))\end{aligned}$$

And if the origins of the coordinate systems $\{A\}$ and $\{B\}$ are coincident, the derivative can be expressed as

$${}^B\dot{\omega}_B = \frac{d}{dt}({}^B\omega_B).$$

For an applied vector ${}^{B}P_Q$ (i.e. the position of point Q in the $\{B\}$ coordinate system) and a rotation matrix ${}^{A}_{B}R$, the time derivative of the position vector of Q, expressed in $\{A\}$, $\frac{d}{dt}({}^{A}P_Q)$ as a function of it's derivative computed in $\{B\}$ is given by [Ref. Sil 91]

$$\begin{aligned}\frac{d}{dt}({}^{A}P_Q) &= \frac{d}{dt}({}^{A}_{B}R\,{}^{B}P_Q + {}^{A}P_{BO}) \\ &= {}^{A}_{B}\dot{R}\,{}^{B}P_Q + {}^{A}_{B}R\frac{d}{dt}({}^{B}P_Q) + {}^{A}V_B \\ &= {}^{A}\Omega_B \times ({}^{A}_{B}R\,{}^{B}P_Q) + {}^{A}_{B}R\,\frac{d}{dt}({}^{B}P_Q) + {}^{A}V_B\end{aligned}$$

Therefore the velocity of the point Q can be expressed in $\{A\}$ as

$${}^{A}V_Q = {}^{A}\Omega_B \times ({}^{A}_{B}R\,{}^{B}P_Q) + {}^{A}_{B}R\,{}^{B}V_Q + {}^{A}V_B$$

or expressing the velocity of Q in the $\{B\}$ coordinate system

$${}^{B}({}^{A}V_Q) = {}^{B}({}^{A}\Omega_B) \times {}^{B}P_Q + {}^{B}V_Q + {}^{B}({}^{A}V_Q)$$

In the case where the origins of $\{A\}$ and $\{B\}$ are coincident then the resulting expression is

$${}^{B}v_Q = {}^{B}\omega_B \times {}^{B}P_Q + {}^{B}V_Q + {}^{B}v_B.$$

C. EQUATIONS USED FOR LINEARIZATION

For any vector equation, $H(c) = a \times b$ where $c = [a\ b]$ and a, b, and c are all vectors, the Taylor series approximation can be written as

$$H(c) = H(c_0) + \frac{\partial H}{\partial c}|_{c=c_0}\delta c + H.O.T.$$

and

$$\frac{\partial H}{\partial c} = \frac{\partial H}{\partial a}|_{a=a_0}\delta a + \frac{\partial H}{\partial b}|_{b=b_0}\delta b$$

Now $\partial H/\partial a$ can be written as

$$\frac{\partial H}{\partial a} = \frac{\partial(a \times b)}{\partial a} = \frac{\partial(-b \times a)}{\partial a} = \frac{\partial(-b)}{\partial a} \times a - b \times \frac{\partial a}{\partial a} = -b\times$$

Using the same relations, $\partial H/\partial b$ can be written

$$\frac{\partial H}{\partial b} = \frac{\partial(-b \times a)}{\partial b} = a \times$$

APPENDIX B: NUMERICAL RESULTS

A. AROD RESULTS

The following results are from the numerical linearization of the kinematic equations of motion for the AROD. Using the trim command, based on an initial $\Theta_0 = \pi/2$, resulted in the state vector and input vector of:

$$x = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1.5700 \\ 0 \end{bmatrix} \text{ and } u = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The linmod command, [a,b,c,d]=linmod('baslnmod',x,u), produced the following linearized system:

$$a = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.5024 \\ 0 & 0 & 0 & 0 & 1.5112 & 0 \end{bmatrix},$$

$$b = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and the c and d matrices were empty, since no outputs were defined.

The following results are for the numerical linearization with gravitational effects added to the 2-3-1 Euler angle kinematic equation model. The trim com-

mand, using the same initial conditions as before, resulted in the following vectors:

[x,u]=trim('gravlmod',x0,[],[],ix,[],[])

$$x = \begin{bmatrix} -0.00 \\ 0.00 \\ -0.00 \\ 0.00 \\ 0.00 \\ 0.00 \\ 0.00 \\ 1.5700 \\ 0.00 \end{bmatrix} \quad \text{and } u = \begin{bmatrix} 32.1740 \\ -0.00 \\ -0.0256 \\ -0.00 \\ 0.00 \\ -0.00 \end{bmatrix}.$$

Linearizing the system, based on the state and input vector found by trimming at the desired conditions resulted in: [a,b,c,d]=linmod('gravlmod',x,u)

$$a = \begin{bmatrix} 0 & 0.00 & 0 & 0 & 0.00 & 0.00 & 0 & -0.0255 & 0.0002 \\ -0.00 & 0 & 0.00 & -0.00 & 0 & 0.00 & 0.0256 & -0.00 & 32.1740 \\ 0.00 & -0.00 & 0 & -0.00 & -0.00 & 0 & -0.00 & -32.1740 & 0 \\ 0 & 0 & 0 & 0 & -0.00 & -0.00 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.00 & 0 & -1.5204 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.00 & 1.5112 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.00 & -0.00 & 0.00 & 0 & 0 & -0.00 \\ 0 & 0 & 0 & 0 & 1.00 & -0.00 & -0.00 & 0 & 0.00 \\ 0 & 0 & 0 & 0 & 0.00 & 1.00 & -0.00 & 0 & 0 \end{bmatrix}$$

and

$$b = \begin{bmatrix} 1.00 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.00 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.00 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.00 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.00 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.00 \end{bmatrix}.$$

Trimming the full EOM model at hover conditions resulted in the following:

$$x = \begin{bmatrix} -0.00 \\ -0.00 \\ 0.00 \\ -0.00 \\ 0.00 \\ -0.00 \\ 0.00 \\ 1.5708 \\ 0.00 \end{bmatrix} \quad \text{and } u = \begin{bmatrix} 0.00 \\ 0.00 \\ 0.1807 \\ 6387.2 \end{bmatrix}.$$

The subsequent linearization for the complete model yielded:

$$a = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0002 & 0.0002 \\ 0.00 & 0 & 0 & 0.00 & 0 & 0.00 & 0.00 & -0.00 & 32.1740 \\ 0 & 0 & 0 & 0.00 & 0 & 0 & -0.00 & -32.1740 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.00 & 0 & -1.5174 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.00 & 1.5082 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.00 & -0.00 & 0 & -0.00 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.00 & -0.00 & 0.00 & 0 & 0.00 \\ 0 & 0 & 0 & 0 & 0 & 1.00 & 0.00 & 0 & 0 \end{bmatrix}$$

and

$$b = \begin{bmatrix} 0 & 0 & 0 & 0.0112 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 24.8194 & 0.0003 \\ -6.6192 & 0 & 0 & -0.00 \\ 0 & -6.5791 & 0 & -0.00 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Computation of the eigenvalues gave:

$$\text{eig(a)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.00 \\ 0 + 1.5128i \\ 0 - 1.5128i \\ 0 \end{bmatrix}.$$

Now consider the quaternion based model Rpm is fixed at 6800 RPM.

To hold the desired vertical attitude, the quaternion states were held fixed:

$$ix = \begin{bmatrix} 7 \\ 8 \\ 9 \\ 10 \end{bmatrix},$$

where the state vector of nominal conditions was:

$$x0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.7071 \\ 0 \\ 0.7071 \\ 0 \end{bmatrix}$$

Trim and linearization of the quaternion–based gravitational model resulted in the following state and input vectors, as well as linearized a and b matrices:

$$x = \begin{bmatrix} -0.00 \\ 0.00 \\ -0.00 \\ 0.00 \\ -0.00 \\ 0.00 \\ 0.7071 \\ -0.00 \\ 0.7071 \\ 0.00 \end{bmatrix}, \quad u = \begin{bmatrix} 32.1740 \\ -0.00 \\ -0.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{bmatrix},$$

$a =$, columns 1–6,

$$\begin{bmatrix} 0 & 0.00 & 0.00 & 0 & 0.00 & 0.00 \\ -0.00 & 0 & 0.00 & -0.00 & 0 & 0.00 \\ -0.00 & -0.00 & 0 & -0.00 & -0.00 & 0 \\ 0 & 0 & 0 & 0 & 0.00 & -0.00 \\ 0 & 0 & 0 & 0.00 & 0 & -1.6051 \\ 0 & 0 & 0 & 0.00 & 1.6062 & 0 \\ 0 & 0 & 0 & 0.00 & -0.3536 & -0.00 \\ 0 & 0 & 0 & 0.3536 & -0.00 & 0.3536 \\ 0 & 0 & 0 & 0.00 & 0.3536 & 0.00 \\ 0 & 0 & 0 & -0.3536 & -0.00 & 0.3536 \end{bmatrix}$$

columns 7–10

$$\begin{bmatrix} -45.5009 & 0 & -45.5009 & 0 \\ 0.00 & -45.5009 & 0.00 & 45.5009 \\ 45.5012 & -0.0003 & -45.5012 & 0.0003 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -0.00 & 0.00 & -0.00 \\ 0.00 & 0 & 0.00 & 0.00 \\ -0.00 & -0.00 & 0 & 0.00 \\ 0.00 & -0.00 & -0.00 & 0 \end{bmatrix},$$

and

$$b = \begin{bmatrix} 1.00 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.00 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.00 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.00 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.00 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.00 \end{bmatrix}.$$

Eigenvalues are computed as:

$$\text{eig(a)} = \begin{bmatrix} -0.00 + 0.00i \\ -0.00 - 0.00i \\ 0.00 \\ 0 \\ -0.00 + 1.6057i \\ -0.00 - 1.6057i \\ -0.00 \\ 0.00 \\ -0.00 + 0.00i \\ -0.00 - 0.00i \end{bmatrix}.$$

B. BLUEBIRD NUMERICAL RESULTS

Results for the numerical trim and linearization of the Bluebird model are given below. The results are from the trim of the kinematic equations of motion for the

Bluebird model are based on the nominal conditions of:

$$x0 = \begin{bmatrix} 72.9954 \\ 0 \\ 6.6757 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.0912 \\ 0 \end{bmatrix}$$

with the states, u, w, and Θ held constant, $ix = [1\ 5\ 8]'$. The trim command, [x,u]=trim('basic',x0,[],[],ix,[],[]) resulted in the following state and input vectors:

$$x = \begin{bmatrix} 72.9954 \\ 0 \\ 6.6757 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.0912 \\ 0 \end{bmatrix}, \text{ and } u = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The numerical linearization of the basic model, [A,B,C,D]=linmod('basic',x,u), resulted in the following A and B matrices:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & -6.6757 & 0 \\ 0 & 0 & 0 & 6.6757 & 0 & -72.9954 \\ 0 & 0 & 0 & 0 & 72.9954 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The C and D matrices were empty since no outputs were defined.

Results for trim and linearization of the kinematic model with gravitational effects added yielded the following state and input vectors:

$$x = \begin{bmatrix} 72.9954 \\ 0.00 \\ 6.6763 \\ 0.00 \\ 0.00 \\ -0.00 \\ -0.00 \\ 0.0912 \\ 0.00 \end{bmatrix}, \text{ and } u = \begin{bmatrix} 2.8772 \\ 0.00 \\ -31.4605 \\ -0.00 \\ 0.3945 \\ -0.00 \end{bmatrix}$$

The numerical linearization of the model resulted in the following A and B matrices:

$A =$

$$\begin{bmatrix} -0.0007 & -0.00 & 0.0079 & 0 & -6.6763 & 0.00 & -0.00 & -32.0451 & 0 \\ 0.00 & 0 & 0.00 & 6.6763 & 0 & -72.9954 & 32.0403 & 0.00 & 0 \\ -0.0001 & 0 & 0.0007 & -0.00 & 72.9954 & 0 & -0.0002 & -2.8773 & 0 \\ -0.00 & 0.0087 & -0.00 & 0 & -0.00 & 0.00 & -0.00 & -0.00 & 0 \\ -0.00 & 0.00 & 0.0005 & 0.00 & 0 & -0.00 & 0.00 & 0.0361 & 0 \\ -0.00 & 0.0010 & 0.00 & -0.00 & -0.00 & 0 & -0.00 & -0.00 & 0 \\ 0 & 0 & 0 & 1.00 & -0.00 & 0.0915 & 0.00 & -0.00 & 0 \\ 0 & 0 & 0 & 0 & 1.00 & 0.00 & 0.00 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.00 & 1.0042 & 0.00 & -0.00 & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Results for trimming the entire model were $A =$

$$\begin{bmatrix} -0.0622 & 0.00 & 0.3431 & 0 & -1.6187 & 0.00 & 0.00 & -32.0416 & 0 \\ 0.00 & -0.3865 & 0.00 & 1.7450 & 0 & -71.6817 & 32.0403 & 0.00 & 0 \\ -0.7558 & 0.00 & -4.7145 & 0.00 & 67.1233 & 0 & -0.0002 & -2.8771 & 0 \\ 0.00 & -0.1455 & 0.00 & -5.3700 & 0.00 & 1.5003 & 0.00 & 0.00 & 0 \\ 0.0155 & 0.00 & -0.1907 & 0.00 & -3.1266 & 0.00 & 0.00 & 0.0362 & 0 \\ 0.00 & 0.1418 & 0.00 & -1.0589 & 0.00 & -0.7970 & 0.00 & 0.00 & 0 \\ 0 & 0 & 0 & 1.00 & 0.00 & 0.0915 & 0.00 & 0.00 & 0 \\ 0 & 0 & 0 & 0 & 1.00 & 0.00 & 0.00 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.00 & 1.0042 & 0.00 & 0 & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} -4.3850 & 0 & 0 & 8.7745 \\ 0.00 & 5.6803 & 0 & 0 \\ -37.8929 & 0 & 0 & 0 \\ 0.00 & 0.6216 & 45.9851 & 0 \\ -21.4821 & 0 & 0.00 & 0 \\ 0.00 & -7.1281 & -6.1465 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Complete model linearized with $\Theta_0 = 0$ instead of $\Theta_0 = \alpha_0$ The initial conditions are now:

$$x0 = \begin{bmatrix} 73.3000 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The state and input vectors obtained from trimming at this state are

$$x = \begin{bmatrix} 73.3000 \\ -0.00 \\ 1.6086 \\ -0.00 \\ -0.00 \\ 0.00 \\ 0.00 \\ -0.00 \\ -0.00 \end{bmatrix} \text{ and } u = \begin{bmatrix} -0.0181 \\ -0.00 \\ 0.00 \\ 0.2336 \end{bmatrix}$$

with the linearized A and B matrices: $A =$

$$\begin{bmatrix} -0.0635 & 0.00 & 0.3277 & 0 & -1.4922 & 0 & -0.00 & -32.1740 & 0 \\ -0.00 & -0.3911 & -0.00 & 1.6086 & 0 & -72.6109 & 32.1740 & -0.00 & 0 \\ -0.7572 & -0.00 & -4.7741 & 0 & 67.9934 & 0 & -0.0002 & -0.0002 & 0 \\ 0.00 & -0.1471 & -0.00 & -5.4414 & -0.00 & 1.5183 & 0.00 & -0.00 & 0 \\ 0.0151 & -0.00 & -0.1933 & 0 & -3.1672 & 0 & 0.00 & -0.00 & 0 \\ 0.00 & 0.1440 & 0.00 & -1.0578 & 0.00 & -0.8114 & 0.00 & -0.00 & 0 \\ 0 & 0 & 0 & 1.00 & 0 & -0.00 & 0 & -0.00 & 0 \\ 0 & 0 & 0 & 0 & 1.00 & -0.00 & -0.00 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.00 & 1.00 & -0.00 & -0.00 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} -4.5835 & 0 & 0 & 8.7745 \\ 0.00 & 5.8282 & 0 & 0 \\ -38.8681 & 0 & 0 & 0 \\ -0.00 & 0.6252 & 47.1717 & 0 \\ -22.0417 & 0 & 0 & 0 \\ -0.00 & -7.3151 & -6.4345 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The eigenvalues for the case where $\Theta_0 = 0$ are

$$eig(A) = \begin{bmatrix} 0 \\ -0.5285 + 3.6346i \\ -0.5285 - 3.6346i \\ -5.6291 \\ -3.9833 + 3.5521i \\ -3.9833 - 3.5521i \\ 0.0420 \\ -0.0191 + 0.4963i \\ -0.0191 - 0.4963i \end{bmatrix}.$$

C. CESSNA 172 RESULTS

The Cessnal 172 model was trimmed and linearized using the state vector

$$x0 = \begin{bmatrix} 219 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

as specified in [Ref. Ros 79]. The states u, w, and Θ were held constant making the term $ix = [1\ 3\ 8]'$.

Only the complete model was linearized, with the results of the trim routine

given as:

$$x = \begin{bmatrix} 219.00 \\ 0.00 \\ -0.0394 \\ -0.00 \\ -0.00 \\ -0.00 \\ -0.00 \\ -0.00 \\ 0.00 \end{bmatrix} \text{ and } u = \begin{bmatrix} 0.0001 \\ 0.00 \\ 0.00 \\ 1.0013 \end{bmatrix}.$$

The linearized model had the following results for the A and B matrices $A =$

$$\begin{bmatrix} -0.0442 & 0.00 & 0.0848 & 0 & 0.0382 & 0 & 0.00 & -32.1740 & 0 \\ 0.00 & -0.1620 & -0.00 & -0.0394 & 0 & -217.2141 & 32.1740 & -0.00 & 0 \\ -0.2916 & -0.00 & -2.1805 & 0 & 212.5399 & 0 & -0.0002 & -0.0002 & 0 \\ 0.00 & -0.1313 & 0.00 & -12.4093 & 0.00 & 2.5342 & -0.00 & -0.00 & 0 \\ 0.0024 & -0.00 & -0.1085 & 0 & -6.0778 & 0 & 0.00 & -0.00 & 0 \\ -0.00 & 0.0462 & 0.00 & -0.3807 & 0.00 & -1.2600 & 0.00 & -0.00 & 0 \\ 0 & 0 & 0 & 1.00 & 0 & -0.00 & 0 & -0.00 & 0 \\ 0 & 0 & 0 & 0 & 1.00 & -0.00 & 0.00 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.00 & 1.00 & -0.00 & -0.00 & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} -6.2509 & 0 & 0 & 3.2255 \\ -0.00 & 19.4571 & 0 & 0 \\ -44.3392 & 0 & 0 & 0 \\ 0.00 & 4.7446 & 57.4954 & 0 \\ -39.4919 & 0 & 0 & 0 \\ -0.00 & -10.2288 & -8.2562 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and the following eigenvalues

$$eig(A) = \begin{bmatrix} 0 \\ -12.4309 \\ -0.6947 + 3.3080i \\ -0.6947 - 3.3080i \\ -4.1303 + 4.3895i \\ -4.1303 - 4.3895i \\ -0.0109 \\ -0.0209 + 0.1794i \\ -0.0209 - 0.1794i \end{bmatrix}.$$

APPENDIX C: PROGRAM LISTINGS

A. AROD MATLAB ROUTINES

1. Main Routine

```
function accel=main(vstate,m,rho,A,R)
%       Function will compute the accelerations due to the
%       gravitational forces, aerodynamic forces and moments,
%       and control forces and moments.
%           The values for S,rho,m,b,and c are used as input
%           arguments to the function call, and are loaded
%           from the workspace. There should be a file,
%           loaddata.m loaded prior to running the
%           simulation.

% define states in terms of the input vector
u=vstate(1);
v=vstate(2);
w=vstate(3);
p=vstate(4);
q=vstate(5);
r=vstate(6);
phi=vstate(7);
theta=vstate(8);
psi=vstate(9);
```

```
%  Define the control inputs
de=vstate(10);
dr=vstate(11);
da=vstate(12);
drpm=vstate(13);


% this subroutine computes linear and angular accelerations
% given angular and linear velocities;
% the input is 6x1 vector = [u v w p q r]'
% the output is feedback part of d/dt [u v w p q r]'
% the output also includes the Euler angle derivatives, based on
% a 2-3-1 transformation, for Ldot, used in the function grav.m.

v = vstate(1:3);
omega = vstate(4:6);
vdot  = -crpr(omega,v);
[Ib,Ir] = inertia;
% compute the angular momentum due to the body's inertia, Ib
Lb = Ib *omega;
% compute the angular momentum due the spinning prop's inertia, Ir
OmegaR=drpm*2*pi/60;                    % angular velocity of the prop, rad/sec
Lr=Ir*[OmegaR;0;0];
temp=Lr+Lb;
omdot = -Ib\crpr(omega,temp);
```

```
vstatedot=[vdot;omdot];

%  Use the Euler angle propagation equations for a 2-3-1 rotation sequence
Ldot=[p/cos(psi)-cos(phi)*sin(psi)/cos(psi)*q+sin(phi)*sin(psi)/cos(psi)*r;
      cos(phi)/cos(psi)*q-sin(phi)/cos(psi)*r;
      sin(phi)*q+cos(phi)*r];

%           Given the vector containing the state derivatives,
%           The function will compute the forces and moments due to
%           the control derivatives, Cfd, where this
%           is dCf/dd.
%           The values for rho,A,R,m are used as input
%           arguments to the function call, and are loaded
%           from the workspace. There should be a file,
%           arod.mat loaded prior to running the
%           simulation.
%
% hover case V=0, dimensionalize the control derivatives based on
% induced velocity through the rotor disk, Vi=sqrt(T/2/A/rho)
%  Calculate the quantities needed for the force
%  calculation.

%  Call a function to return the stability derivatives wrt to moments
%  Could put a call to a lookup table here
%  Syntax--Z = TABLE2(TAB,X0,Y0) or Y = TABLE1(TAB,X0)
```

```
Cfd=getcfd;
% Define the derivatives
Clda=Cfd(4,3);
Cmde=Cfd(5,1);
Cndr=Cfd(6,2);
% Calculate the Force due to Cfd derivatives
% No Aerodynamic Forces in a Hover
D=0;
Y=0;
L=0;

% calculate the force due to thrust in body coordinates
% USING THRUST VS. RPM FROM BOB STONEY'S TEST RUNS

T=0.0297*drpm-104.7;
Vi=sqrt(T/2/A/rho)
qbar=.5*rho*Vi^2;   % Vi is induced velocity, not forward speed
Fout=[D;Y;L];
Fout=(Fout+[T;0;0])/m;

% Calculate the Moment due to Cfd derivatives
% ltr  relates the duct swirl to the moment, l, produced by thrust
ltr=-0.0542*T-0.9138;
l=qbar*A*R*(Clda*da)+ltr;
m=qbar*A*R*(Cmde*de);
n=qbar*A*R*(Cndr*dr);
```

```
Nout=Ib\[l;m;n];
FNcfx=[Fout;Nout];

%           Given the vector containing the state derivatives,
%           and euler angles, the function will compute
%           the forces due to gravity acting on the aircraft.
%
%
%  Calculate gravitational force, based on a 2-3-1 Euler angle
%  rotation for position of the aircraft. Rotation matrix is Ru2body
%  Z for {U} is positive down and X for {B} is straight up.

FNgrav=32.174*[-sin(theta)*cos(psi);
                sin(theta)*sin(psi)*cos(phi)+sin(phi)*cos(theta);
                cos(phi)*cos(theta)-sin(theta)*sin(psi)*sin(phi);
               0;0;0];

%   Sum up the normalized forces and moments to feed back into the integrator

vstatedot=[vstatedot+FNcfx+FNgrav;Ldot];
accel=vstatedot;
```

2. Supporting Subroutines

```
function y = crpr(omega,x)
```

```
% this subroutine computes the crossproduct of omega and x:
% y = omega X x
p = omega(1);
q = omega(2);
r = omega(3);
t = [0 -r q;
     r 0 -p;
    -q p 0];
y = t * x;


function Fgrav=grav(x)
% GRAV will compute the gravitational force
% acting on the body, in body coordinates

g=x(1);
phi=x(2);
theta=x(3);
psi=x(4);
Fgrav=g*[-sin(theta)*cos(psi);
          sin(theta)*sin(psi)*cos(phi)+sin(phi)*cos(theta);
          cos(phi)*cos(theta)-sin(theta)*sin(psi)*sin(phi)];


function [ib,ir] = inertia
% this subroutine creates inertia matrices  called ib
```

```
% and ir for the body and rotor inertia, respectively.
% Arod hover case
ixx  =  1.2312;
iyy = 3.9584;
izz = 3.9825;
ixz = 0;
irx=.00898;
iry=.0045;
irz=.0045;
ib = [ixx 0 -ixz;0 iyy 0;-ixz 0 izz];
ir = [irx 0 0;0 iry 0; 0 0 irz];
```

3. Data and Initialization Subroutines

```
% load bluebird data
A=3.14;                 % Area of rotor disk, ft^2
Vt=712.09;              % Rotor tip speed, rad/sec
m=2.6419;               % Mass, slugs
R=1;                    % Radius of Rotor Blade, ft
rho=.002377;            % Air density, slug/ft^3
Uo=0;                   % Nominal Velocity, ft/sec

%  Initial Euler Angle Orientation, radians
Phi=0;
Theta=1.57;             % Same as Steady State alpha
Psi=0;
Lo=[Phi;Theta;Psi];
```

```
%  Initial Conditions to determine the Aircraft
%  Linear and Rotational Velocity States
%  u,v,w are computed from Uo, alpha, and beta
%  p,q,r are assumed as zero.
alpha=Theta;
beta=0;
Xo=[Uo;alpha;beta];

%  Returns Initial Conditions for the main
%  integrator in the rigid body EOM block
i0=init_var(Lo,Xo);

function Cfd=getcfd
% Cfd=getcfd will return values for
% The stability derivatives for D,Y,L,l,m,and n
% due to the control inputs.
% format for data is;
% [CDde CDdr CDda
%  CYde ...
%  CLde ...
%  Clde Cldr Clda
%  Cmde ...
%  Cnde ...  Cnda
%  Data is non-dimensionalized by using induced velocity
%  Vi=sqrt(T/2/A/rho)
```

```
% for arod hover case
  Cfd=[    0    0    0;
           0    0    0;
           0    0    0
           0    0  1.438;
        -1.233 0     0;
           0  -1.233  0];


function i0=init_var(Lo,Xo)
%  INIT_VAR(X) will initialize the integrators
%  initial states, i0, given the initial
%  conditions desired.
%  Required initial conditions are the Euler
%  angle orientation, total velocity, Uo, initial
%  AOA, and sideslip angle, beta.
%  Lo=[phi;theta;psi]'
%  Xo=[Uo;alpha;beta]'
%  All body rotation rates are assumed to be zero


%  Initialize the states u,v,w,p,q,r
Uo=Xo(1);
alpha=Xo(2);
beta=Xo(3);
```

```
Ca=cos(alpha);
Sa=sin(alpha);
CB=cos(beta);
SB=sin(beta);

Rwb=[Ca*CB -Ca*SB -Sa;SB CB 0;Sa*CB -Sa*SB Ca];
v0=Rwb*[Uo;0;0];
w0=[0;0;0];
i0=[v0;w0;Lo];



function Qo=initq(lambda)
%  Function initQ will return values for
%  the quaternion DCM based on a given
%  set of Euler angles.
%  Set for a Euler 3-2-1 rotation
%  Q(1)=B0
%  Q(2)=B1
%  Q(3)=B2
%  Q(4)=B3
phi=lambda(1);
theta=lambda(2);
psi=lambda(3);

Qo(1)=.5*sqrt(1+cos(psi)*cos(theta)+sin(psi)*sin(theta)*sin(phi)...
```

```
        +cos(psi)*cos(phi)+cos(theta)*cos(phi));
Qo(2)=1/4/Qo(1)*(cos(theta)*sin(phi)-sin(psi)*sin(theta)*cos(phi)...
        +cos(psi)*sin(phi));
Qo(3)=1/4/Qo(1)*(cos(psi)*sin(theta)*cos(phi)+sin(psi)*sin(phi)...
        +sin(theta));
Qo(4)=1/4/Qo(1)*(sin(psi)*cos(theta)-cos(psi)*sin(theta)*sin(phi)...
        +sin(psi)*cos(phi));
Qo=Qo';
```

B. BLUEBIRD MATLAB ROUTINES

1. Main Routine

```
function accel=main(vstate,rho,b,c,S,m,Xo)
%       Function will compute the accelerations due to the
%       gravitational forces, aerodynamic forces and moments,
%       and control forces and moments.
%           The values for S,rho,m,b,and c are used as input
%           arguments to the function call, and are loaded
%           from the workspace. There should be a file,
%           loaddata.m loaded prior to running the
%           simulation.

% define states in terms of the input vector
u=vstate(1);
v=vstate(2);
w=vstate(3);
```

```
p=vstate(4);
q=vstate(5);
r=vstate(6);
phi=vstate(7);
theta=vstate(8);
psi=vstate(9);

%  Define the control inputs
de=vstate(10);
dr=vstate(11);
da=vstate(12);
dt=vstate(13);

% calculate the aerodynamic terms
Vt=sqrt(u^2+v^2+w^2);
qbar=.5*rho*(Vt)^2;

Ib=inertia;

% wind to body transformation
Rwb=rw2b(vstate,Vt);



%
%
%-------------------------------------------------------------------------
```

```
% CHI will compute the left hand side of the state
% equation. This is the term dependant on dCf/dxdot.
% Now calculate the S matrix that non-dimensionalizes
% the moments. Also includes the correction for Lift and Drag
% acting in the direction opposite to the positive coordinate
% direction.
% Get the stability derivatives for forces and moments
Cfxdot=getcfxd;
CLad=Cfxdot(3);
Cmad=Cfxdot(5);

% Twb is a intermediate step
Twb=[eye(3)/m zeros(3);zeros(3) inv(Ib)]*[Rwb zeros(3);zeros(3) Rwb];

% calculate the M2 matrix to allow use of the states, rather than
% the normalized states. Only accounts for the alpha dot variables
% since the beta dot terms are not ordinarily computed.

%M2=[0 0 1 0 0 0]*c/2/Vt^2;
%Sprime=diag([-S S -S S*b S*c S*b]);
%  To save some math here, the product of Sprime, Cfxdot, and M2 is:
PROD=[0 0 0 0 0 0;0 0 0 0 0 0;0 0 -S*CLad*c/2/Vt^2 0 0 0;
      0 0 0 0 0 0;0 0 S*c*Cmad*c/2/Vt^2 0 0 0;0 0 0 0 0 0];
% Calculate chi
chi1=(eye(6)-Twb*qbar*PROD);
%
```

```
%
%-------------------------------------------------------------------------

%           Given the vector containing the state derivatives,
%           and euler angles, the function will compute
%           the forces due to gravity acting on the aircraft.

%  Calculate gravitational force, based on a 3-2-1 Euler angle
%  rotation for position of the aircraft. Rotation matrix is Ru2body

Fgrav=32.174*[-sin(theta);
               cos(theta)*sin(phi);
               cos(theta)*cos(phi)];

% Premultiply by the Chi^-1 term from the left hand side
FNgrav=chi1\[Fgrav;0;0;0];
%
%-------------------------------------------------------------------------
%


% Cfx(u)  Given the vector containing the state derivatives,
%           The function will compute the forces and moments due to
%           the stability derivatives, Cfx', where this
%           is dCf/dx.
%
```

```
% Call a function to return the stability derivatives wrt to moments
% Could put a call to a lookup table here
% Syntax--Z = TABLE2(TAB,X0,Y0) or Y = TABLE1(TAB,X0)

Cfx=getcfx;
CDu=Cfx(1,1);  CDa=Cfx(1,3);
CYb=Cfx(2,2);  CYp=Cfx(2,4);  CYr=Cfx(2,6);
CLu=Cfx(3,1);  CLa=Cfx(3,3);  CLq=Cfx(3,5);
Clb=Cfx(4,2);  Clp=Cfx(4,4);  Clr=Cfx(4,6);
Cmu=Cfx(5,1);  Cma=Cfx(5,3);  Cmq=Cfx(5,5);
Cnb=Cfx(6,2);  Cnp=Cfx(6,4);  Cnr=Cfx(6,6);

ss=getcf0;
CD0=ss(1);
CL0=ss(3);
Cm0=ss(5);

Cfd=getcfd;
% Define the derivatives
CDde=Cfd(1,1);
CYdr=Cfd(2,2);  CYda=Cfd(2,3);
CLde=Cfd(3,1);
Cldr=Cfd(4,2);  Clda=Cfd(4,3);
Cmde=Cfd(5,1);
Cndr=Cfd(6,2);  Cnda=Cfd(6,3);
```

```
%  Calculate the Force due to Cfx' derivatives
%  And the control derivatives
%  in the wind coordinates

D=-S*qbar/m*(CD0+CDa*w/Vt+CDde*de);
Y=S*qbar/m*(CYb*v/Vt+CYr*r*b/2/Vt+CYdr*dr+CYda*da);
L=-S*qbar/m*(CL0+CLa*w/Vt+CLq*q*c/2/Vt+CLde*de);

%  calculate the force due to thrust in body coordinates
%  THRUST IS ESTIMATED, BASED ON 4.0 HP, PROP EFFICENCY=.65

T=15;
Xt=T/m*dt;

Fout=Rwb*([D;Y;L]);
Fout=Fout+[Xt;0;0];

%  Calculate the Moment due to Cfx' derivatives
%  And the control derivatives
l=qbar*S*b*(Clb/Vt*v+Clp*b/2/Vt*p+Clr*b/2/Vt*r+Cldr*dr+Clda*da);
m=qbar*S*c*(Cm0+Cma/Vt*w+Cmq/Vt*c/2*q+Cmde*de);
n=qbar*S*b*(Cnb/Vt*v+Cnp*b/2/Vt*p+Cnr*b/2/Vt*r+Cndr*dr+Cnda*da);

Nout=Ib\(Rwb*[l;m;n]);
```

```
% Premultiply by the Chi^-1 term from the left hand side
FNcfx=chi1\[Fout;Nout];
%
%-------------------------------------------------------------------------------



% this subroutine computes linear and angular accelerations
% given angular and linear velocities;
% the input is 6x1 vector = [u v w p q r]'
% the output is feedback part of d/dt [u v w p q r]'
% the output also includes the Euler angle derivatives
% Ldot, used in the function grav.m.

v = vstate(1:3);
omega = vstate(4:6);
vdot  = -crpr(omega,v);
temp = Ib*omega;
omdot = -Ib\crpr(omega,temp);

vstatedot=chi1\[vdot;omdot];

%  Use the Euler angle propagation equations
Ldot=[p+sin(phi)*tan(theta)*q+cos(phi)*tan(theta)*r;
                cos(phi)*q-sin(phi)*r;
                sin(phi)/cos(theta)*q+cos(phi)/cos(theta)*r];
```

```
vstatedot=[vstatedot+FNcfx+FNgrav;Ldot];
accel=vstatedot;
```

2. Supporting Subroutines

```
function y = crpr(omega,x)
% this subroutine computes the crossproduct of omega and x:
% y = omega X x
p = omega(1);
q = omega(2);
r = omega(3);
t = [0 -r q;
     r 0 -p;
    -q p 0];
y = t * x;
```

```
function Fgrav=grav(x)
% GRAV will compute the gravitational force
% acting on the body, in body coordinates

g=x(1);
phi=x(2);
theta=x(3);
```

```
Fgrav=g*[-sin(theta);
          cos(theta)*sin(phi);
          cos(theta)*cos(phi)];


function Ib = inertia
% this subroutine creates inertia matrix  called Ib
% for the Bluebird test aircraft.
% All units are slug-ft^2
Ix=10.0;
Iy=16.12;
Iz=7.97;
Ib=[Ix 0 0;0 Iy 0;0 0 Iz];


function [Rwb]=Rw2b(x,Vt)
% RWIND2BODY Rotation matrix for wind to body coordinate transformations.

        Ca=x(1)/Vt;
        Sa=x(3)/Vt;
        CB=x(1)/Vt;
        SB=x(2)/Vt;

Rwb=[Ca*CB -Ca*SB -Sa;SB CB 0;Sa*CB -Sa*SB Ca];
```

3. Data and Initialization Subroutines

```
% load bluebird data
S=22.38;                % Planform Area, ft^2
Uo=73.3;                % Nominal Velocity, ft/sec
m=1.7095;               % Mass, slugs
b=12.42;                % Span, ft
c=1.802;                % Average Chord, ft
rho=.002377;            % Air density, slug/ft^3

%  Initial Euler Angle Orientation, radians
Phi=0;
Theta=.0912;            % Same as Steady State alpha
Psi=0;
Lo=[Phi;Theta;Psi];

%  Initial Conditions to determine the Aircraft
%  Linear and Rotational Velocity States
%  u,v,w are computed from Uo, alpha, and beta
%  p,q,r are assumed as zero.
alpha=Theta;
beta=0;
Xo=[Uo;alpha;beta];

%  Returns Initial Conditions for the main
%  integrator in the rigid body EOM block
i0=init_var(Lo,Xo)
```

```
function i0=init_var(Lo,Xo)
%  INIT_VAR(X) will initialize the integrators
%  initial states, i0, given the initial
%  conditions desired.
%  Required initial conditions are the Euler
%  angle orientation, total velocity, Uo, initial
%  AOA, and sideslip angle, beta.
%  Lo=[phi;theta;psi]'
%  Xo=[Uo;alpha;beta]'
%  All body rotation rates are assumed to be zero

%  Initialize the Euler angle DCM

%  Initialize the states u,v,w,p,q,r
Uo=Xo(1);
alpha=Xo(2);
beta=Xo(3);

Ca=cos(alpha);
Sa=sin(alpha);
CB=cos(beta);
SB=sin(beta);

Rwb=[Ca*CB -Ca*SB -Sa;SB CB 0;Sa*CB -Sa*SB Ca];
v0=Rwb*[Uo;0;0];
```

```
w0=[0;0;0];
i0=[v0;w0;Lo];


function Cf0=getcf0
% Cf0=getcf0 will return values for
% the nominal values for coefficients
% format of input is [CD0 CY0 CL0 Cl0 Cm0 Cn0]';

Cf0=[0.03 0 0.3 0 0 0]';


function Cfd=getcfd
% Cfd=getCfd_F(n) will return values for
% The stability derivatives for D,Y,and L
% due to the control inputs.
% format for data is;
% [CDde CDdr CDda
%  CYde ...
%  CLde ...
%  Clde Cldr Clda
%  Cmde ...
%  Cnde ...  Cnda]

% For the test aircraft Bluebird
% Derivatives from DATCOM
```

```
  Cfd=[   .065   0      0;
           0    .0697   0;
          .472   0      0
           0   .0028  .265;
         -1.41  0      0;
           0  -.0329  -.0347];


function Cfx=getcfx
% Cfx=getcfx(n) will return values for
% The stability derivatives for D,Y,and L
% due to the state vector.
% format of data:
%[CDu CDb CDa CDp CDq CDr;
% CYu ...
% CLu ...
% Clu ...
% Cmu ...
% Cnu ...                ]

% For the test aircraft Bluebird
% Derivatives from DATCOM

  Cfx=[0    0    .188     0    0      0;
       0  -0.31   0       0    0    .0973;
       0    0    4.22     0   3.94    0;
```

```
        0  -.0597    0  -.363    0      .1;
        0     0  -1.163     0  -11.77    0;
        0    .0487    0  -.0481  0     -.0452];

function Cfxdot=getcfxd
% Cfxdot=getcfxd(n) will return values for
% The stability derivatives for D,Y,and L
% due to the state vector. Beta dot terms are ignored
% since they are not normally determined.
% format is:
%[ CDadot
%  CYad
%  CLad
%  Clad
%  Cmad
%  Cnad ]

Cfxdot=[0;0;1.32;0;-4.7;0];
```

REFERENCES

[Siu 91] Siuru, W.D., *Planes Without Pilots: Advances in Unmanned Flight,* TAB Books, Blue Ridge Summit, PA, 1991.

[Wh 87] White, J.E., and Phelan, J.R., "Stability Augmentation for a Free Flying Ducted Fan," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, Aug. 1987, pp 896–904.

[MCG 87] Jennings Jr, J.F., "Why the Corps Needs Robots," *Marine Corps Gazette,* Vol. 5, No. 1987, pp 36–39.

[Sa 89] Not Attributed, *Sandia Science News* Vol. 1, 1989

[Kre 92] Kress, G.A., "Preliminary Development of a VTOL Unmanned Air Vehicle for the Close-Range Mission." *Master's Thesis, Department of Aeronautics*, Naval Postgraduate School, Monterey, CA, 1992.

[Sto 93] Stoney, R.B., "Design, Fabrication, and Test of a Vertical Attitude Take–Off and Landing Unmanned Air Vehicle," *Engineer's Thesis, Department of Aeronautics*, Naval Postgraduate School, Monterey, CA, June 1993.

[We 88] Weir, R.J., "Aerodynamic Design Considerations for Free–Flying Ducted Propellor," *Proceedings of the 1988 Atmospheric Flight Mechanics Conference,* AIAA, Washington, D.C., Aug. 1988, pp. 720-731.

[DOD 92] "DoD Unmanned Aerial Vehicle Master Plan," Department of Defense, Washington, D.C., 1992.

[Wh 91] White, J.E., and Phelan, J.R., "Stability Augmentation and Control Decoupling for the Airborne Remotely Operated Device," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No.1, 1991, pp 176-183.

[Sil 91] Silvestre, C.J., "Modeling and Control of Underwater Vehicles," *Master's Thesis, Department of Electrical Engineering*, Instituto Superior Tecnico, Lisbon, Portugal, 1991.

[Cra 86] Craig, J.J., *Introduction to Robotics Mechanics and Control,* Addison-Wesley, New York, 1986.

[Ju 92] Junkins, J.L., *An Introduction to Dynamics and Control of Flexible Structures,* AIAA, Washington D.C., 1992.

[Sch 92] Schmidt, L.V., *Class Notes for AE3340,* U.S. Naval Postgraduate School, Monterey, CA. 1992.

[Ka 83] Kane, T.R., Likins, P.W., Levinson, D.A., *Spacecraft Dynamics,* McGraw–Hill, New York, 1983.

[Mo 84] Morton, H.S., "A Formulation of Rigid-Body Rotational Dynamics Based on Generalized Angular Momentum Variables Corresponding to the Euler Parameters," *Proceedings of the AIAA/AAS*, Seattle, WA, August 1984.

[Ro 58] Robinson, A.C., "On the Use of Quaternions in Simulation of Rigid Body Motion," *WADC Technical Report 58-17,* Wright Air Development Center, December, 1958.

[Whi 59] Whittaker, E.T., *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies,* Cambridge Univ. Press, 4th Edition, 1959.

[Gre 88] Greenwood, D.T., *Principles of Dynamics, 2nd Ed.,* Prentice–Hall, Englewood Cliffs, N.J., 1988.

[Ros 79] Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls,* Roskam Aviation and Engineering corp, Ottawa, KS, 1979

[Th 89] Thompson, C.M., "Aircraft Equations of Motion and Forming Linear Models," Boeing Document D6–54972, Boeing Commercial Airplane Company, Seattle, Washington, 1989.

[USAF 60] *USAF STABILITY AND CONTROL HANDBOOK,* Wright Air Development Division, United States Air Force, Wright Patterson AFB McGregor and Werner, Inc., Dayton, OH, 1960.

[Pro 90] Prouty, R.W., *Helocopter Performance, Stability, and Control,* Robert E. Krieger, Malabar, Florida, 1990.

[WAT 93] Watson Industries, *Techical Specifications for IMU–600D* Watson Industries, Eau–Claire, WI, 1993.

[Bro 64] Broxmeyer, C., *Inertial Navigation Systems,* McGraw–Hill, New York, 1964.

[St 88] Strum, R.D., and Kirk, D.E., *First Principles of Discrete Systems and Digigal Signal Processing,* Addison–Wesley, New York, 1988.

[Sp 89] Spong, Vydia, Sagas., *Robot Dynamics and Control,* Wiley, New York, 1989.

INITIAL DISTRIBUTION LIST

		No. of Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Commandant of the Marine Corps Code TE06 Headquarters, U.S. Marine Corps Washington, D.C. 20380-0001	1
3.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
4.	Dr. Isaac I. Kaminer Department of Aeronautics and Astronautics, Code AA/Ka Naval Postgraduate School Monterey, California 93943-5000	5
5.	Dr. Richard W. Howard Department of Aeronautics and Astronautics, Code AA/Ho Naval Postgraduate School Monterey, California 93943-5000	2
6.	Chairman Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, California 93943-5000	2
7.	Capt. David R. Kuechenmeister 1995 Skidmore Circle Lawrenceville, Georgia 30244	2

432 709

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

RES
STSC, guidelines for
\STSC.
route to: RESERVE

Mahr, Randolph L.

ID:3Z768000191530

GAYLORD S

AMERICAN BINDERY
2
6-94

DUDLEY KNOX LIBRARY
3 2768 00019153 0